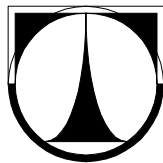


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií



BAKALÁŘSKÁ PRÁCE

Liberec 2007

Jan Křivka

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: Informatika a logistika

Trojrozměrná vizualizace materiálových textilních struktur pomocí virtuální reality

Three-Dimensional Visualisation of Textile Material Structures Using Virtual Reality

Bakalářská práce

Autor:

Jan Křivka

Vedoucí práce:

Prof. RNDr. Ing. M. Košek, CSc.

Konzultant:

Ing. T. Mikolanda

V Liberci 9. 5. 2007

Zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Anotace

Existuje mnoho systémů pro 3D vizualizaci textilních struktur. Většina z nich používá polygonální reprezentaci, která vytváří povrch struktury z polygonů. Navržený přístup využívá moderní prvky virtuální reality. Za použití speciálních uzlů mohou být komplikované objekty jako jsou textilní vlákna velmi dobře modelována. Výhodou je, že uživatel si volí pouze parametry vláken respektive parametry textilních vazeb a systém sám vytvoří trojrozměrný model. Program byl napsán v programovacím prostředí Delphi 5.0, využívá přednosti jazyka VRML, což je jedinečný jazyk pro reprezentaci virtuální reality. Výstupem programu je 3D model tkaniny pro jazyk VRML obsahující všechny výhody tohoto jazyka, jako je například procházení skrz strukturu.

Abstract

There are many systems for 3D visualisation of textile structures. Most of them approximate the surface of the structure by polygons. Proposed approach uses modern elements of virtual reality. Complicated objects like textile fibres can be modelled very easily using special nodes. An advantage of this is that the user chooses fibre parameters or more precisely textile weave parameters only and the system itself creates three-dimensional model. The program was made in the Delphi 5.0 programming environment and uses advantages of the VRML language, which is a unique language for virtual reality representation. Output of the program is 3D fabric model for VRML language that contains all the advantages of this language, including moving through the structure.

Obsah

Úvod.....	9
1. Počítačová grafika.....	10
1.1 Způsoby reprezentace.....	10
1.1.1 Analytická reprezentace.....	10
1.1.2 Polygonální reprezentace.....	11
1.1.3 Optimalizace – zmenšování počtu polygonů.....	12
1.2 Způsoby vyjádření parametrických ploch	12
1.2.1 Bézierovy plochy.....	12
1.2.2 B-spline plochy.....	13
1.2.3 Plochy NURBS	14
1.3 Promítání.....	14
1.4 Světlo.....	14
1.5 Stínování.....	15
1.5.1 Konstantní stínování.....	16
1.5.2 Gouradovo stínování.....	16
1.5.3 Phongovo stínování.....	16
1.6 Řešení viditelnosti – paměť hloubky (z-buffer).....	16
1.7 Globální zobrazovací metody.....	17
1.7.1 Sledování paprsku.....	17
1.7.2 Radiozita.....	17
2. Virtuální realita.....	19
2.1 Stereoskopické pohledy.....	19
3. VRML – Jazyk pro virtuální realitu.....	20
3.1 Hierarchie a uzly.....	21
3.2 Události.....	24
3.3 Světelné zdroje.....	25
3.3.1 DirectionalLight.....	25

3.3.2	PointLight.....	26
3.3.3	SpotLight.....	26
3.4	Obecná tělesa.....	26
3.4.1	Uzel Extrusion.....	27
4.	Nastupující standardy.....	31
4.1	X3D – Formát pro ukládání 3D scén.....	31
4.2	Ajax – Technologie pro vývoj webových aplikací.....	31
4.3	Ajax3D – Třetí rozměr na internetu.....	33
5.	Základní vazby tkanin.....	34
5.1	Plátnová vazba.....	34
5.2	Keprová vazba.....	34
5.3	Atlasová vazba.....	34
6.	Realizace.....	35
7.	Typické výsledky.....	39
8.	Diskuse.....	41
	Závěr.....	42
	Použitá literatura.....	43
Příloha 1:	Projekty obdobného zaměření – Práce Ing. Tomáše Mikolandy.....	45
Příloha 2:	Projekty obdobného zaměření – Práce Ing. Maroše Tunáka.....	48

Seznam obrázků a tabulek

Obrázek 1:	Bézierova křivka.....	13
Obrázek 2:	spektrum viditelného světla.....	15
Obrázek 3:	3D modely s nastavením různých hodnot parametru <i>specularColor</i>	23
Obrázek 4:	3D modely s nastavením různých hodnot parametru <i>transparency</i>	24
Obrázek 5:	3D model pohárku.....	28
Obrázek 6:	3D model tkaniny.....	28
Obrázek 7:	3D model kapky.....	29
Obrázek 8:	plátňová vazba.....	34
Obrázek 9:	keprová vazba.....	34
Obrázek 10:	atlasová vazba.....	34
Obrázek 11:	model plátna s kruhovým průřezem vláken.....	36
Obrázek 12:	model kepru s průřezem vláken ve tvaru elipsy.....	36
Obrázek 13:	grafické rozhraní s popisem zhotoveného programu.....	37
Obrázek 14:	výsledný program zobrazující model tkaniny.....	39
Obrázek 15:	3D model třívazného kepru s kruhovým průřezem vláken a 3D model sedmivazného atlasu s kruhovým průřezem vláken.....	39
Obrázek 16:	3D model plátna s kruhovým průřezem vláken a 3D model třívazného kepru s obdélníkovým průřezem.....	40
Obrázek 17:	3D model pětivazebného kepru s obdélníkovým průřezem vláken a 3D model sedmivazebného kepru s eliptickým průřezem.....	40
Obrázek 18:	3D model pětivazebného atlasu s eliptickým průřezem vláken a 3D model sedmivazebného atlasu s eliptickým průřezem.....	40
Obrázek 19 :	model tkaniny 1.....	45
Obrázek 20:	model tkaniny 2.....	46
Obrázek 21:	model tkaniny 3.....	46
Obrázek 22:	model tkaniny 4.....	47
Obrázek 23:	model tkaniny 5.....	47

Obrázek 24:	plátnová vazba.....	48
Obrázek 25:	třívazný osnovní kepr levého směru.....	48
Obrázek 26:	třívazný osnovní kepr pravého směru.....	48
Obrázek 27:	třívazný útkový kepr levého směru.....	48
Obrázek 28:	třívazný útkový kepr pravého směru.....	49
Obrázek 29:	pětivazný osnovní kepr levého směru.....	49
Obrázek 30:	sedmivazný útkový kepr levého směru.....	49
Obrázek 31:	pětivazný osnovní atlas.....	49
Obrázek 32:	pětivazný útkový atlas.....	49
Obrázek 33:	sedmivazný osnovní atlas.....	49
Obrázek 34:	sedmivazný útkový atlas.....	50
Obrázek 35:	devítivazný útkový atlas.....	50
Tabulka 1:	základní geometrická tělesa.....	21
Tabulka 2:	uzly <i>Transform</i> a <i>Shape</i>	22
Tabulka 3:	parametry uzlu <i>Material</i>	23
Tabulka 4:	parametry uzlu <i>Extrusion</i>	28
Tabulka 5:	výpis textového souboru reprezentující model kapky.....	29
Tabulka 6:	volitelné parametry zhotoveného programu	37

Úvod

Textilní objekty či struktury podle velikosti a užití dělíme na tři typy, přičemž typický rozměr se liší o několik řádů. Jsou to:

1. Vlákná, jejich průměr bývá několik μm . Mluvíme o mikrostruktuře.
2. Nitě, příze, pletence apod., splétané z tisíců vláken. Typický rozměr jsou desetiny mm. Jedná se o mesostrukturu.
3. Textilie vytvářené z nití s typickým rozměrem desetiny metru. Jde o makrostruktury.

Na užitnou hodnotu textilie má vliv zejména struktura vytvářená z nití, podle předchozího dělení tedy mesostruktura. Dochází k tzv. synergickému efektu – výsledné vlastnosti textilie nejsou pouhým součtem vlastností jednotlivých komponent, nití, ale v důsledku vazby přistupují k očekávaným parametrům i další důležité a významné vlastnosti a parametry. Není proto divu, že textilní mesostruktura a vazby jsou intenzivně zkoumány. Jedním z důležitých prostředků výzkumu je i jejich zjednodušené zobrazení, modelování.

Vhodným nástrojem modelování textilních struktur je počítačová grafika, která umožňuje vytvářet prostorové trojrozměrné modely, tj. 3D modely. Existují složité komerční systémy pro přesné zobrazení všech možných textilních struktur, např. systém WiseTex, o němž se zmiňují např. práce [5, 6]. Dále jsou to jednodušší modely struktury založené na univerzálních obecných grafických systémech pro 3D zobrazení, viz např. systém v práci [7] vyvinutý na Technické univerzitě v Liberci (TUL). Na Internetu existují mnohé další různě složité systémy.

Poměrně malá pozornost byla věnována jiné významné možnosti modelování, kterou je virtuální realita. Na rozdíl od univerzálních grafických systémů umožňují jazyky virtuální reality udělat mnoho práce za uživatele. Použití virtuální reality (jazyk VRML – Virtual Reality Modeling Language) je např. v práci [5, 6]. V tomto případě je však povrch vlákna¹ modelován pomocí malých trojúhelníků, takže možností virtuální reality je využito málo. Jako ideální se pro model vlákna hodí uzel *Extrusion* jazyka VRML. Proto se tato práce soustředila na modelování základních textilních struktur pomocí tohoto uzlu.

¹ V dalším textu budeme pro označení prvku mesostruktury používat termín vlákno místo odborně správného termínu nit' příze. Tento termín není vhodný, ale v aplikaci počítačové grafiky je vžitý.

1. Počítačová grafika

Počítačová grafika je obor informatiky a slouží k reprezentaci informací do vizuální podoby. Informace mohou být zobrazovány dvourozměrně (2D) nebo trojrozměrně (3D). S vývojem výpočetní techniky a počítačové grafiky se možnosti grafického vyjádření informací značně rozšířily. Tento trend přispívá k efektivnější a snazší práci uživatele s informacemi v elektronické podobě.

1.1 Způsoby reprezentace

V počítačové grafice lze modely objektů reprezentovat mnoha různými způsoby. Nejčastěji používanou reprezentací v trojrozměrné počítačové grafice je v současné době reprezentace hraniční (*boundary representation, B-rep*), jak se uvádí v literatuře [2]. Existují dva nejčastější způsoby vymezení hranice tělesa. V polygonální (ploškové) reprezentaci jsou oblé plochy tělesa aproximovány elementárními ploškami, nejčastěji trojúhelníky. Druhým způsobem je analytická reprezentace popisující těleso matematickou funkcí – obvykle nějakou hladkou plochou. Oba způsoby mají svá specifika a nacházejí svá uplatnění v různých oblastech počítačové grafiky.

1.1.1 Analytická reprezentace

Analytické vyjádření hraničně reprezentovaných těles se opírá o poměrně rozsáhlou teorii křivek a ploch. Dnes se v počítačové grafice používají převážně parametrické plochy, ale stále častěji jsou aplikovány i implicitní plochy. Charakteristickou vlastností analytického vyjádření ploch je jeho přesnost. Nejčastěji se využívá tam, kde záleží na přesném vyjádření těles, jako jsou CAD systémy, architektura, fyzikální simulace a v mnoha dalších. Díky analytickému vyjádření ploch lze velmi snadno měřit povrch, případně objem těles, která jsou takto reprezentována. Další předností analytického vyjádření je i to, že se neobjevují nepřesnosti při změně měřítka takovýchto objektů. Na rozdíl od polygonální reprezentace lze analytické plochy efektivněji editovat, měnit jejich tvar, provádět s nimi operace jako je jejich rozdělení, průnik atp. Analyticky vyjádřené funkce jsou jednoduše diferencovatelné a tudíž objekty jimi popsané se snadno navazují. V analytické formě bývá obtížné vyjádřit

složitý objekt jako celek, proto některé druhy analytické reprezentace vytvářejí složitější celky z částí jednodušších, u křivek ze segmentů a u ploch z plátů.

Přímé zobrazení analytických ploch je výpočetně náročné, zvláště pokud jsou kladeny vysoké nároky na kvalitu zobrazení. Pak je nejvýhodnější zobrazovat je metodou sledování paprsku (*ray tracing*). Tam, kde je důležitější rychlost zobrazení před matematickou přesností modelu, se používá reprezentace plošková, která se pak zpracovává běžnými technikami, nejčastěji Gouraudovým stínováním a pamětí hloubky. Převod z analytické reprezentace do polygonální je snadný a analytický model je navíc z hlediska objemu dat úspornější. Jelikož algoritmy pro získání polygonální reprezentace z reprezentace analytické jsou velmi rychlé, není nutno udržovat v paměti počítače obě reprezentace. Díky tomu lze polygonální reprezentaci vždy vygenerovat podle potřeby aplikace, například při změně modelu.

1.1.2 Polygonální reprezentace

Polygonální (plošková) reprezentace je nejvíce používanou reprezentací objektů v počítačové grafice. Základním prvkem této reprezentace je trojúhelník, i když většina standardů pro zobrazování prostorových dat (OpenGL, Direct 3D) podporuje i čtyřúhelníky a mnohoúhelníky. U mnohoúhelníků je však při zobrazování nutné určovat jejich konvexitu, což je výpočetně náročnější než u trojúhelníků. Zásadní výhodou polygonální reprezentace je to, že technické vybavení počítačů přímo podporuje zobrazování takto vyjádřených modelů. Většina operací nad objekty reprezentovanými polygonálně v konečném důsledku vede na lineární interpolaci, která je technicky realizovatelná velice snadno. Polygonální reprezentace je velmi efektivní v časově kritických aplikacích, jako je virtuální realita, herní nebo filmový průmysl. Výpočty průsečíku paprsku s trojúhelníkem jsou poměrně rychlé, a proto se v metodě sledování paprsku místo výpočtu průsečíku se složitým objektem tento objekt převádí na polygonální reprezentaci a výpočty průsečíků se provádějí s ní.

Avšak pro modelování je polygonální reprezentace značně nevýhodná. Z tohoto důvodu modelovací programy obvykle pracují s reprezentací jinou (nejčastěji s NURBS, která je popsána dále) a síť trojúhelníků generují až na svém výstupu. Příkladem tohoto přístupu jsou trojrozměrné modely používané ve virtuální realitě. Naprostá většina těchto modelů je vytvářena jako NURBS a síť trojúhelníků je z nich vygenerována jednou. Výstupní síť může být generována ve více variantách lišících se

svou složitostí a tedy i velikostí detailů (*level of details*). Mezi nevýhody polygonální reprezentace patří především nesnadné mapování textur. Dále je nevýhodou tzv. geometrický alias. Tento nežadoucí jev se projevuje při změně měřítka. Původně spojitě těleso, jehož dva uzly jsou reprezentovány dvěma oddělenými údaji, se při změně měřítka může opticky roztrhnout. K tomu dochází numerickou chybou vlivem konečné přesnosti zobrazení čísel v paměti počítače.

Polygon je nejčastěji reprezentován pouze svými vrcholy (*vertex*). Jejich pořadí a tudíž i pořadí jeho hran (*edge*) pak určuje orientaci celého polygonu. Vrchol obvykle neobsahuje pouze své souřadnice v trojrozměrném prostoru, ale nese také informace o normálovém vektoru, který je nezbytný především při osvětlování, dále údaje o barvě, průhlednosti, případně materiálu, ze kterého se objekt skládá.

1.1.3 Optimalizace - zmenšování počtu polygonů

Hlavním důvodem ke snižování počtu trojúhelníků jsou reálné možnosti počítačů při provádění náročných výpočtů a omezení přenosu dat po síti. Optimalizace povrchu objektu je proto velmi důležitou součástí grafických systémů. Je důležité minimalizovat přenos dat jak po sběrnících počítače, tak po síti, ale i mezi pevným diskem a operační pamětí. Optimalizace lze rozdělit do dvou skupin:

1. Redukce počtu trojúhelníků, přičemž se nezmění přesnost popisu.
2. Redukce počtu trojúhelníků při zhoršení kvality reprezentace. Toho se dosáhne zanedbáním méně důležitých detailů.

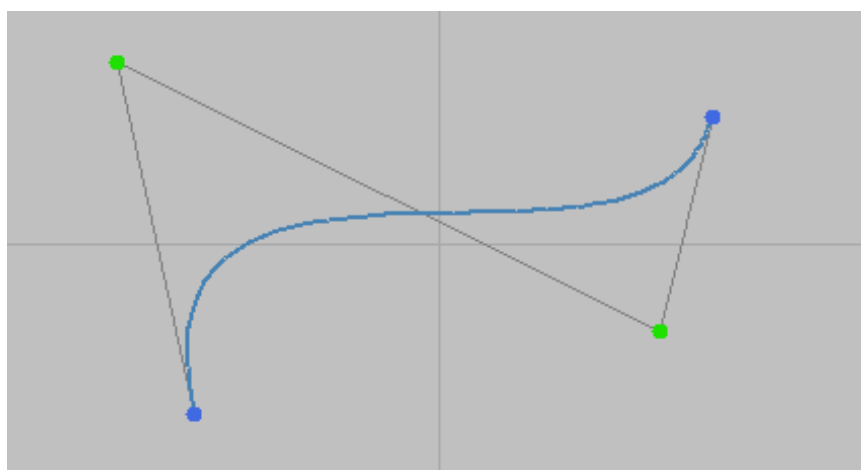
1.2 Způsoby vyjádření parametrických ploch

Parametrické plochy lze vyjádřit více způsoby [2], jedná se hlavně o Bézierovy plochy, B-spline plochy a o NURBS. Jednotlivé způsoby jsou popsány dále.

1.2.1 Bézierovy plochy

Jednodušší modelovací programy používají Bézierovy plochy pro reprezentaci těles. Bézierovy plochy mají tu výhodu, že lze jednoduše dopočítat průsečík s paprskem, jsou snadno diferencovatelné a dobře modelovatelné [2]. Nevýhodou je, že neumožňují vyjádřit kuželosečky. Podmínky hladkého navázání Bézierových ploch, které jsou nutné při jejich plátování, vedou k silnému omezení modelovacích možností. Také Bézierovy

plochy nejsou invariantní k perspektivnímu promítání (viz níže). V počítačové grafice jsou Bézierovy křivky a plochy pro svou jednoduchost velice populární. Avšak nemožnost vyjádření kuželoseček a nízká míra manipulovatelnosti znemožňují nasazení v systémech, kde je zapotřebí velká přesnost a široké spektrum tvarů. Pokud je potřeba generovat perspektivně promítnutou plochu, musíme perspektivně promítat každý její bod, nestačí pouze promítnout řídicí body této plochy a plochu poté generovat běžným způsobem. Z těchto důvodů se Bézierovy plochy používají především v jednodušších modelovacích systémech. Jako dvourozměrná ilustrace je na obrázku 1 Bézierova křivka, která je daná počátečním a koncovým bodem. Tvar křivky určují dva řídicí body.



Obrázek 1: Bézierova křivka

1.2.2 B-spline plochy

B-spline plochy se velice snadno navazují a jsou proto pro modelování daleko výhodnější. V dvourozměrném případě jsou to v podstatě čáry vytvářené dříve křivítkem. B-spline plochy se, stejně jako plochy Bézierovy, zadávají sítí řídicích bodů. Avšak není nutné omezovat některé jejich řídicí body vnějšími podmínkami jako v případě Bézierových ploch. Změnou jediného řídicího bodu se změní tvar vždy pouze těch plátů, které jsou tímto bodem určeny.

1.2.3 Plochy NURBS

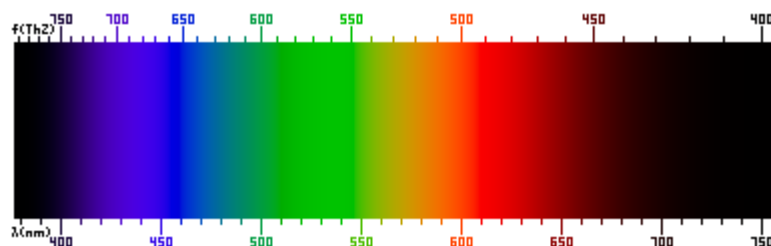
NURBS (*non uniform rational B-spline*) jsou zobecněním B-spline ploch a jsou v dnešní době standardem v geometrickém modelování. Umožňují definování jak volně tvarovatelných ploch, tak kuželoseček, a díky nim i takových geometrických objektů jako je koule, válec či krychle. Mezi výhody patří též to, že plochy NURBS jsou (stejně jako Bézierovy plochy) invariantní k lineárním transformacím a k perspektivní projekci. Plochu, jejíž každý bod zobrazujeme v perspektivní projekci, což je výpočetně velice náročné, můžeme získat tak, že nejprve podrobíme perspektivní projekci její řídicí body a potom vygenerujeme plochu bez nutnosti dále transformovat její body. Nevýhodou může být poměrně složitý matematický popis.

1.3 Promítání

Namodelované trojrozměrné objekty se zobrazují na dvojrozměrné zobrazovací zařízení. Převedení trojrozměrných objektů do dvojrozměrné podoby vede ke ztrátě informace a je realizováno promítáním. Dnes promítáním, nebo též projekcí rozumíme transformaci z n -rozměrného do m -rozměrného prostoru pro $m < n$. V počítačové grafice se používá rovnoběžné a středové promítání na jednu průmětnu. Průmětna je plocha v prostoru, na kterou dopadají promítací paprsky a v místě dopadu vytvářejí průmět (obraz v rovině). Během promítání následuje další zpracování dat, jedná se o nalezení zakrytých a viditelných částí objektů, vyhodnocení jejich barvy a nanesení. Poté lze objekt korektně zobrazit.

1.4 Světlo

Přítomnost světla je základní podmínkou pro to, aby naše oko mohlo vnímat okolní svět. Světelné paprsky přinášejí informace tím, jak dopadají na sítnici oka. Z těchto informací si vytváříme představu o barvě, umístění a tvaru objektů v prostoru. Lidské oko vnímá světlo v úzkém elektromagnetickém spektru od $4,3 \cdot 10^{14}$ Hz (700nm), odpovídající červené barvě, až po $7,5 \cdot 10^{14}$ Hz (400nm) odpovídající barvě fialové.



Obrázek 2: spektrum viditelného světla

Světelný zdroj, jakým je slunce nebo žárovka, vysílá paprsky všech frekvencí v daném pásmu, které se tak skládají ve výsledné bílé světlo. Toto světlo se nazývá achromatické. Dopadne-li bílé světlo na objekt, jsou některé frekvence povrchem objektu odraženy a některé jsou pohlceny. Kombinace frekvencí přítomných v odraženém světle vytváří to, co vnímáme jako barvu objektu. Pro popis charakteristik světla jsou užitečné kromě frekvence i další vlastnosti:

Jas, světlost (*intensity, lightness*), případně svítivost (*luminosity*) odpovídá intenzitě světla. Čím vyšší je intenzita světla, tím se jeví zdroj jasnější.

Sytost (*saturation*) udává čistotu barvy světla. Je tím vyšší, čím užší je frekvenční spektrum světla.

Odstín, barevný tón (*hue, brightness*) určuje velikost achromatické složky ve světle s určitou dominantní frekvencí.

1.5 Stínování

Stínování je proces, který provádí výpočet osvětlovacího modelu. Výpočet je vykonáván pouze pro několik bodů na povrchu tělesa, ostatní jsou odvozené od těch vypočtených. Následně se vypočítávají barevné odstíny všech zobrazovaných bodů. Pomocí stínování (*shading*) lze odlišit případné křivosti a zaoblení ploch, a tím tak docílit přirozeného vzhledu prostorových objektů. Toho je docíleno i přesto, že řada výpočtů týkajících se zpracování světla byla zjednodušena či vynechána. Některé druhy stínování umožňují opticky vyhladit povrchy, které jsou modelovány sítí rovinných plošek, takže přestanou být znatelné drobné hraniční zlomy. Stínování se nezabývá nalezením vržených stínů (*shadows*), ty zpracovávají globální osvětlovací techniky.

1.5.1 Konstantní stínování

Konstantní stínování je velmi rychlé a jednoduché. Je používáno tam, kde je třeba docílit vysoké rychlosti zobrazení. Předpokládá, že každá plocha má jen jedinou normálu. Podle normály je vypočítán jeden barevný odstín, který je po rasterizaci plochy přiřazen všem jejím pixelům. Používá se jen pro zobrazování rovinných ploch.

1.5.2 Gouraudovo stínování

Metoda byla navržena Henri Gouraudem a je vhodná pro stínování těles, jejichž povrch je tvořen množinou rovinných plošek. Tato metoda vypočítává barevné odstíny vnitřních bodů dané plošky pomocí bilineární interpolace. Pro činnost algoritmu je důležitá znalost barvy všech vrcholů zpracovávané plochy. Gouraudova metoda zajišťuje plynulé stínování křivých povrchů tak, že aproximace povrchů ploškami není zřetelná. Přesto neposkytuje zcela věrný obraz reálných objektů.

1.5.3 Phongovo stínování

Metoda vypočítává z normálových vektorů ve vrcholech stínované plochy nejen barevné odstíny ve vrcholech (jako Gouraudova metoda), ale určuje také normálové vektory ve vnitřních bodech plochy bilineární interpolací. Phongovo stínování je založeno na interpolaci normálových vektorů. Z normálových vektorů je vyhodnocením osvětlovacího modelu určen odstín barvy každého pixelu. Jelikož je osvětlovací model vyhodnocován v každém bodě plochy, tak ve srovnání s Gouraudovým stínováním výrazně vzrostou časové nároky algoritmu.

1.6 Řešení viditelnosti – paměť hloubky (z-buffer)

Tato metoda patří k nejznámějším a také nejefektivnějším [2]. Dosahuje vysoké rychlosti zpracování a to i přesto, že klade poměrně vysoké nároky na paměť. Je používána jako standardní prostředek pro řešení viditelnosti v naprosté většině grafických akceleratorů. Základem metody je použití paměti hloubky (z-buffer), která tvoří dvourozměrné pole. Každá položka paměti hloubky obsahuje souřadnici z toho bodu, který leží nejbližší k pozorovateli a jehož průmět leží v odpovídajícím pixelu rastru.

1.7 Globální zobrazovací metody

V reálném modelu existuje více objektů, které vzájemně ovlivňují svůj vzhled. Odražené světlo nebo vržený stín jednoho objektu může ovlivnit objekt jiný, který se také nachází ve scéně. V počítačové grafice existují dvě zobrazovací metody, které se snaží postihnout vzájemné vztahy mezi objekty. Jedná se o sledování paprsku (*ray tracing*) a o radiozitu.

1.7.1 Sledování paprsku

Světelné paprsky se šíří od světelných zdrojů různými směry. Jsou paprsky, které zasáhnou povrch objektů ve scéně a jsou paprsky, které opustí prostor scény. Podle optických vlastností těchto objektů se paprsky odrážejí nebo lomí, a ovlivňují osvětlení v dalších místech scény. V počítačové grafice se proces šíření světla modeluje obráceným způsobem, proto se také této metodě někdy říká zpětné sledování paprsku. Při vykreslování se sleduje dráha světelného paprsku od každého bodu průmětny směrem ke zdroji světla. Během cesty paprsek naráží do objektů, které mění jeho barvu a odráží ho novým směrem. Kdyby byl sledován paprsek obráceně, skončila by většina paprsků mimo průmětnu a počítaly by se zbytečně.

Existují dvě základní varianty. První je sledování paprsku prvního řádu. Při něm se zobrazuje pouze bod na povrchu nejbližšího tělesa zasaženého paprskem. V tomto bodě se stanoví barva pomocí osvětlovacího modelu.

Druhou metodou je sledování paprsku vyššího řádu. Sledování paprsku nekončí po nalezení prvního tělesa, ale pokračuje sledováním dalších paprsků, odvozených podle odrazivosti a průhlednosti nalezeného tělesa. Tato metoda dokáže zobrazit na povrchu tělesa zrcadlové obrazy jiných těles. Je také schopna zpracovat vržené stíny.

1.7.2 Radiozita

Tato metoda umožňuje věrněji simulovat reálný svět z fyzikálního hlediska. Radiozita využívá poznatky z oblasti výpočtů tepelného záření pro výpočet osvětlovacího modelu. Základní radiozitní algoritmus vychází ze zákona zachování energie a předpokládá, že přenos světelného záření mezi objekty probíhá v energeticky uzavřené scéně a že není ovlivněn prostředím a všechny objekty jsou neprůhledné. Objekty jsou popsány ploškovou hraniční reprezentací.

Postup při zobrazování scény metodou radiozity má dvě části. Nejprve je vyhodnoceno šíření světla ze světelných zdrojů a jeho odrazy na povrchu těles. Výsledkem tohoto výpočtu je ohodnocení ploch koeficienty, které vyjadřují množství difúzně odraženého světla pro každou plochu. Tyto koeficienty nezávisí na poloze pozorovatele, jsou pouze vlastností scény. Proto lze ve druhé části metody použít libovolný zobrazovací algoritmus řešící viditelnost scény a zobrazovat scénu z různých pohledů bez nutnosti nových výpočtů difúzního odrazu světla.

2. Virtuální realita

Volně přeloženo, virtuální realita je zdánlivá skutečnost [2]. Charakteristickou vlastností je interakce uživatele s virtuálním světem generovaným pomocí výpočetní techniky. Důležitým předpokladem pro efektivní využívání virtuální reality je pohyb a interakce v reálném čase. Další, avšak neméně důležitou vlastností, je schopnost nabídnout pocit, že se uživatel nachází uvnitř reálného světa. Existují různé druhy periférií, které zefektivňují práci ve virtuálním světě a umocňují pocit reálnosti. Jedná se o stereoskopické brýle, zvukovou aparaturu poskytující prostorový zvuk, snímače detekující prostorovou polohu uživatele, rukavice s hmatovými senzory aj. Virtuální svět nebývá statickou scénou, uživatel může s jednotlivými objekty manipulovat.

2.1 Stereoskopické pohledy

Pocit, že se člověk nachází v trojrozměrném prostoru, je nejvíce ovlivněn zrakem. Dvě zornice lidských očí jsou od sebe vzdáleny přibližně 7 cm a do mozku dodávají mírně rozdílné pohledy. Ty jsou posléze automaticky skládány a z jemných rozdílů mezi nimi je odvozena hloubka (vzdálenost). Přesvědčivost systémů pro virtuální realitu je často založena právě na vytvoření dvou tzv. stereoskopických pohledů. Jde o obrazy vzniklé perspektivním promítáním ze dvou bodů ležících vedle sebe ve vzdálenosti 7 cm. Vypočtené stereoskopické pohledy je třeba zobrazit pro každé oko zvlášť. V helmách pro virtuální realitu je to o něco lehčí, protože před každým okem je samostatná obrazovka. Je také možné použít běžnou velkou obrazovku doplněnou synchronizovanými zatmívacími brýlemi. Brýle zatmívají střídavě levý a pravý zorník tak, jak je na obrazovku střídavě vykreslován obraz pro pravé, resp. levé oko.

3. VRML – Jazyk pro virtuální realitu

Jazyk VRML (Virtual Reality Modeling Language) je určen pro popis obsahu virtuálních světů a jejich chování. Je oficiálně definován mezinárodní normou ISO/IEC 14772-1:1997. Virtuální světy zapisuje do souborů v textovém tvaru. Jazyk VRML podporuje především ploškovou hraniční reprezentaci objektů. Je to objektově orientovaný jazyk, jehož prvky mají tři základní vlastnosti objektů:

1. Zapouzdřenost (*encapsulation*)
2. Mnohotvarost (*polymorphism*)
3. Dědičnost (*inheritance*)

Scénu organizuje do stromové struktury, která zabezpečuje dědění vlastností [2]. Jazyk umožňuje spolupráci s dalšími programovacími jazyky (Java, Javascript) i aktivaci programů, typicky prohlížečů stránek WWW. Scénu lze vytvářet z prvků umístěných jak lokálně v souborech, tak i na vzdálených počítačích v síti internet. Jazyk obsahuje prostředky pro popis animace objektů a interakce s uživatelem.

Plochy objektů lze mapovat texturami různých formátů, dokonce i animovanou sekvencí ve formátu MPEG-1. Soubory se statickými a pohyblivými obrazy přitom mohou být umístěny kdekoliv na síti WWW. Vzhledem k tomu, že spojení na internetu je bez záruk a údaje jsou proměnlivé, je ve formátu VRML zavedena možnost zápisu posloupnosti hypertextových odkazů namísto jediného odkazu. Systém prezentující VRML scénu se snaží získat potřebné údaje postupným prohledáváním těchto odkazů. Stromová struktura scény je vytvářena pomocí tzv. skupinových uzlů, jejichž následníci (potomci) vždy sdílejí stejné vlastnosti. Nejčastěji je používán uzel transform, který určuje polohu svých potomků. Mezi skupinové uzly patří i billboard, jehož potomci se natáčejí vůči pozorovateli.

Interakce ve virtuálním světě VRML jsou řízené buď počítačem nebo přímo uživatelem. Mezi uzly zajišťující uživatelem řízené akce patří skupina tzv. manipulátorů. Dovolují uživateli, aby měnil polohu virtuálních objektů, potomků manipulátoru. Počítačem řízené akce jsou spojeny s existencí senzorů. Jsou to uzly, které detekují rozličné děje ve virtuálním světě a zasílají o nich zprávy dalším uzlům. Senzory jsou schopny zaznamenat, zda uživatel uchopil objekt, zda při svém průchodu

virtuálním světem do objektu narazil nebo do určitého prostoru vstoupil, případně zda určitý objekt vidí ve svém zorném úhlu.

3.1 Hierarchie a uzly

Standard VRML 97 definuje množinu objektů pro vytváření trojrozměrných scén. Tyto objekty jsou definovány v takzvaných uzlech (*nodes*) a jsou uspořádány do hierarchické struktury zvané strom (*tree*). Strom je neorientovaný graf, který je souvislý a nemá cykly (acyklický). Počet jeho hran je o jednu menší než počet vrcholů. Nejjednodušší strom tvoří jediný uzel. Tento uzel se nazývá kořen (*root*). Každý uzel může být propojen s několika dalšími uzly, ty jsou pak jeho potomky (*children*). Naopak z pohledu potomků je předchozí uzel jejich rodičem (*parent*). Celou scénu pak tvoří graf scény (*scene graph*), což je les stromů, tento graf obsahuje alespoň jeden strom. Každý strom v lese stromů je samostatný, na ostatních stromech nezávislý objekt. Uzly v jednotlivých stromech spolu komunikují prostřednictvím událostí. Podstatnou vlastností stromu je určení pořadí mezi jednotlivými uzly. Uzel, který je rodič, může určovat vlastnost všech uzlů, které tvoří množinu všech jeho potomků.

V tabulce 1 jsou uvedeny uzly definující základní geometrická tělesa, tedy základní stavební prvky virtuálních světů.

Těleso	uzel	parametr	iniciální hodnot	význam
koule	Sphere { }	radius	1	poloměr
kvádr	Box { }	size	2 2 2	délky stran x, y a z
kužel	Cone { }	bottomRadius height	1 2	poloměr podstavy výška
válec	Cylinder { }	radius height	1 2	poloměr výška

Tabulka 1: základní geometrická tělesa

Uzly *Transform* a *Shape* patří k nejpodstatnějším uzlům jazyka VRML. Díky uzlu *Transform* můžeme nastavovat důležité vlastnosti těles, jako je například poloha tělesa ve scéně. Uzel *Shape* umožňuje shrnovat objekty, a tím i jejich vlastnosti, do jednoho celku. Základní parametry uzlů *Transform* a *Shape* jsou uvedeny v tabulce 2.

Funkce uzlu	uzel	parametr	Iniciální hodnota	Význam
umístění	<i>Transform</i> { }	<i>Scale</i> <i>Rotation</i> <i>Translation</i> <i>Children</i>	1 1 1 0 0 1 0 0 0 0 Prázdný uzel	měřítko osa a úhel natočení posunutí seznam potomků
svázání vlastností	<i>Shape</i> { }	<i>Geometry</i> <i>appearance</i>	Prázdný uzel Prázdný uzel	tvár povrchu vzhled povrchu

Tabulka 2: uzly *Transform* a *Shape*

Do parametru *appearance* se umísťuje pouze jediný možný uzel, který se jmenuje *Appearance*, ten dovolu je definovat dva možné vzhledy povrchu těles:

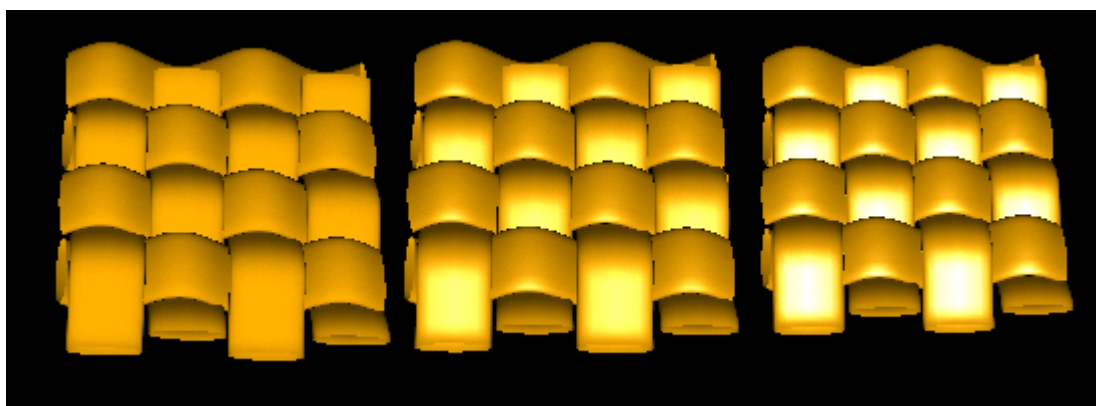
1. barvu (parametr *material*),
2. texturu (parametry *texture* a *textureTransform*).

Parametr *material* slouží k definování barvy a povrchu tělesa a parametr *texture* je určený pro pokrytí tělesa texturou. Do parametrů uzlu *Appearance* se opět přiřazují pouze uzly, jejichž jména jsou odvozena ze jmen parametrů. Například do parametru *material* lze přiřadit pouze uzel *Material*. Nejdůležitější parametry uzlu *Material* jsou uvedeny v tabulce 3, jak je uvedeno v práci [1].

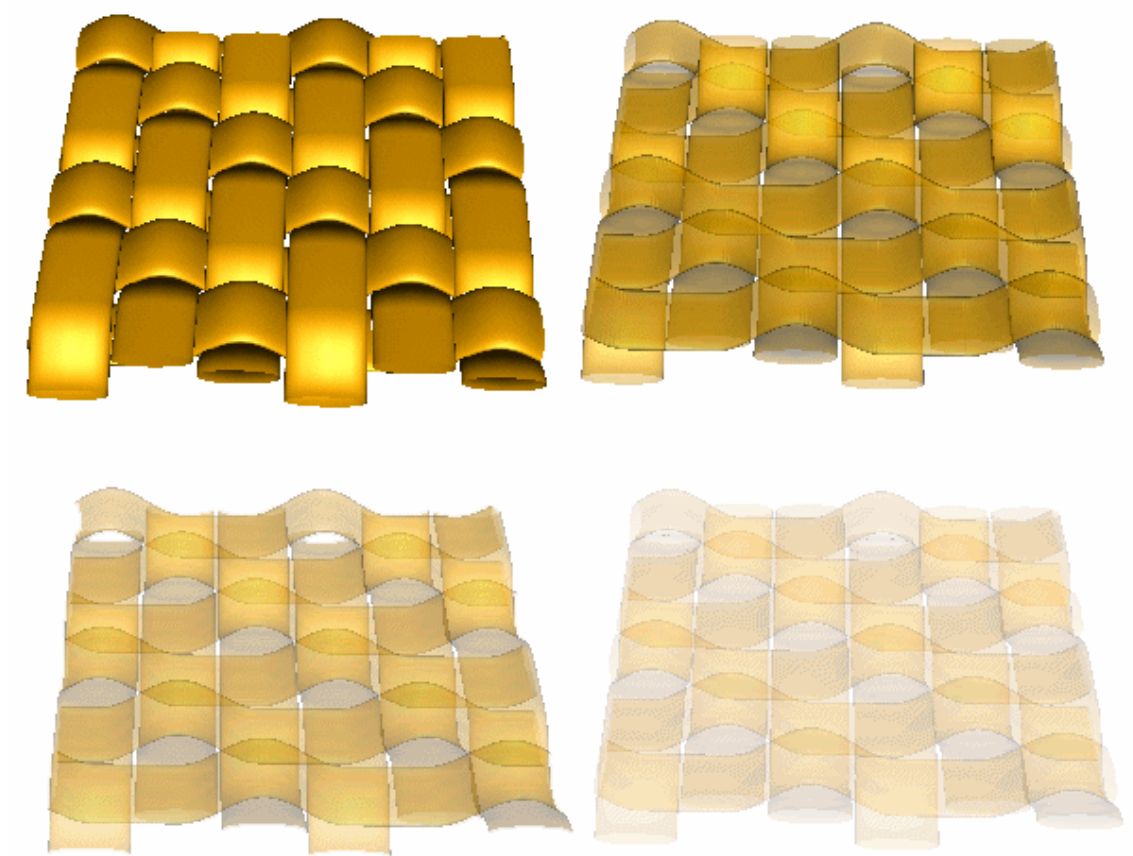
parametr	výchozí hodnota	Význam
diffuseColor	0.8 0.8 0.8	základní složení barvy povrchu ve složkách RGB
ambientIntensity	0.2	jak je barva povrchu zesvětlována celkovým jasnem prostoru
specularColor	0 0 0	jakou barvu dopadajícího světla povrch odráží
shininess	0.2	ostrost odrazu pro předchozí parametr
emissiveColor	0 0 0	fluorescenční (vyzařovaná) barva povrchu
transparency	0	průhlednost (1 = absolutně průhledné)

Tabulka 3: parametry uzlu *Material*

Příklady vlivu parametru *specularColor* uzlu *Material* (pro modelovanou textilní strukturu) demonstruje obrázek 3. Podobně je na obrázku 4 znázorněn vliv parametru *transparency* uzlu *Material*. Oba obrázky se vztahují k textilním strukturám modelovaným v rámci této práce.



Obrázek 3: 3D modely s nastavením různých hodnot parametru *specularColor*
(zleva 0 0 0, 0.5 0.5 0.5, 1 1 1)



Obrázek 4: 3D modely s nastavením různých hodnot parametru *transparency*
(postupně 0, 0.6, 0.8, 0.9)

3.2 Události

Od verze 2.0 je ve VRML obsažena skupina uzlů, které mohou vysílat události (events). Pomocí těchto událostí může uzel reagovat na změny prostředí nebo na podněty od uživatele. Událost vyslaná z některého uzlu zpravidla generuje řadu dalších událostí nebo sama mění strukturu grafu.

Události zpracovávají speciální uzly nazvané *Script*. Událost, která dorazí do tohoto uzlu, v něm způsobí spuštění funkce.

3.3 Světelné zdroje

Ve VRML 97 jsou k dispozici tři uzly, které definují tři různé druhy světelného zdroje.

DirectionalLight je směrové světlo s rovnoběžnými paprsky.

PointLight je bodové světlo s paprsky vycházejícími z jediného bodu.

SpotLight je kuželové světlo s paprsky vycházejícími z bodu, které jsou ohraničeny kuželem.

Jak se uvádí v práci [3], jednotlivé světelné zdroje lze ve virtuálním světě libovolně kombinovat, zdroje jsou na sobě nezávislé. Pokud na určitou část objektu dopadá světlo z několika různých světelných zdrojů, výsledná intenzita osvětlení této části bude součtem všech intenzit.

Všechny tři uzly obsahují atributy *intensity* (intenzita), *color* (barva), *ambientIntensity* (intenzita okolního světla) a *on* (zapnuto). Intenzita představuje přímý světelný jas vycházející ze světelného zdroje. Hodnoty intenzit se mohou pohybovat v rozmezí 0,0 (nesvítí) až do hodnoty 1,0 (maximální jas). Atribut *on* je typu SBoolean a může nabývat dvou hodnot. Hodnota TRUE znamená, že světlo svítí. Druhá hodnota FALSE představuje vypnuté světlo.

Světla *PointLight* a *SpotLight* osvětlují všechny objekty, které se nacházejí v jejich světelných prostorech. *PointLight* má tento prostor ve tvaru koule se středem ve zdroji světla. Světelný kužel představuje světelný prostor pro světlo *SpotLight*. Světlo *DirectionalLight* osvětluje pouze objekty, které jsou na stejné nebo nižší úrovni v grafu scény.

3.3.1 DirectionalLight

Uzel *DirectionalLight* (přímé světlo) definuje světelný zdroj, který vyzařuje rovnoběžné paprsky ve směru vektoru *direction* (směr). Tomuto druhu světla se říká vzdálené (paralelní) světlo a používá se k simulaci slunečního svitu. Atribut *direction* (směr) definuje směr, jakým se bude šířit světlo ze světelného zdroje. S tímto vektorem jsou rovnoběžné všechny světelné paprsky, které se budou šířit prostorem tak dlouho, dokud nedopadnou na nějaký předmět.

3.3.2 PointLight

Uzel *PointLight* definuje bod, ve kterém se nachází světelný zdroj. Tento zdroj světla má charakter bodového světla a jeho paprsky se šíří všemi směry. To, v jakém bodě se světlo nachází, určuje atribut *location*. Jeho implicitní hodnota je 0 0 0, což znamená, že uzel *PointLight* bez atributu *location* bude ležet ve středu souřadné soustavy. Jeho konečná poloha se může ještě změnit transformacemi zděděnými po rodičích. Světelná intenzita dopadající na určitou plochu klesá se zvětšující se vzdáleností plochy od světelného zdroje. To, jakým způsobem bude klesat, určuje atribut *attenuation* (útlum). Bodové světlo bude osvětlovat pouze objekty, které leží od bodu určeného hodnotou atributu *location* ve vzdálenosti menší, než je hodnota atributu *radius* (poloměr). Všechny transformace typu *scale* ovlivní hodnotu atributu *radius*, ostatní transformace zase ovlivní hodnotu atributu *location*.

3.3.3 SpotLight

Pomocí tohoto uzlu lze do scény umístit světelný zdroj, který vyzařuje světlo určitým směrem a jehož paprsky jsou ohraničeny neviditelným kuželem. Uzel *SpotLight* osvětluje pouze ta geometrická tělesa, která leží ve směru vektoru *direction* a protínají tento neviditelný kužel. Stejně jako u uzlu *PointLight*, atribut *attenuation* určuje způsob klesání intenzity dopadajícího světla na plochu. Všechny transformace typu *scale* ovlivní hodnotu atributu *radius*, ostatní transformace, jako je posunutí, zase ovlivní hodnotu atributu *location*.

3.4 Obecná tělesa

Pro popis obecnějších tvarů nabízí jazyk VRML možnost použití dalších uzlů. Pro dvourozměrné objekty to jsou:

Množina čar (*IndexedLineSet*),

Množina bodů (*PointSet*),

Nápis (*Text*).

Pro vytvoření trojrozměrných objektů jsou k dispozici uzly:

Množina ploch (*IndexedFaceSet*),

Opláštění (*Extrusion*),

Výšková mapa (*ElevationGrid*).

Oproti uzlům pro základní tělesa mají tyto uzly složitější strukturu, lze jimi však modelovat téměř jakékoliv tvary.

Velice používaným uzlem je Množina ploch (*IndexedFaceSet*). Tento uzel umožňuje definovat obecná tělesa nebo libovolně zakřivenou plochu tím, že popíše plošky, které daný objekt pokrývají. Plochu popisuje zapsáním indexů vrcholů, nikoliv konkrétním uvedením jejich souřadnic. V případě, že je jeden vrchol sdílen více plochami, jeho souřadnice stačí zapsat jen jednou a zapamatovat si jeho pořadí v seznamu vrcholů. Tento způsob zápisu vrcholů vede k úspoře paměti. Uzel Množina ploch (*IndexedFaceSet*) lze použít v podstatě na jakékoliv těleso včetně textilních struktur. Dalším uzlem, kterým lze modelovat textilní struktury, je uzel Opláštění (*Extrusion*).

3.4.1 Uzel Extrusion

Tento uzel je určen pro popis objektů, které lze vždy charakterizovat rovinným profilem, který je posouván a otáčen v prostoru. Profil a trajektorie jsou určeny pouze jednou lomenou čarou. Takovéto objekty se nazývají translační nebo rotační tělesa. Uzel *Extrusion* má tři základní parametry: *CrossSection*, *Spine* and *Scale*. Hodnoty těchto parametrů udávají výsledný tvar objektu. V parametru *CrossSection* se nastavuje posloupnost bodů v rovině určující profil, takzvanou obrysovou křivku. V parametru *Spine* se nastavuje posloupnost bodů v prostoru určující trajektorii obrysových křivek. Parametr *Scale* obsahuje seznam kladných měřítek pro každou polohu obrysu. Profil, který je určený parametrem *CrossSection*, je opakovaně posouván v prostoru podle poloh v parametru *Spine* a je škálován podle hodnot v parametru *Scale*. Bližší popis je v tabulce 4, uvedeno podle publikace [1].

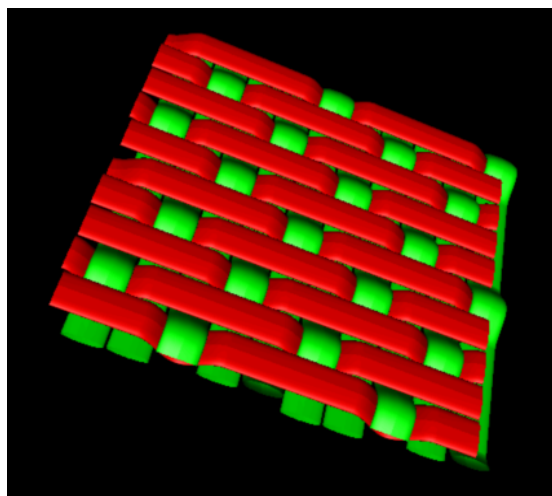
Parametr	Iniciální hodnota	Význam
crossSection	[1 1, 1 -1, -1 -1, -1 1, 1 1]	Posloupnost bodů v rovině určující profil, tj. obrysovou křivku
spine	[0 0 0, 0 1 0]	Posloupnost bodů v prostoru určující trajektorii
scale	[1 1]	Seznam kladných měřítek pro každou další polohu obrysu
orientation	[0 0 1 0]	Seznam otočení pro každou další polohu obrysu
beginCap	TRUE	Povolení vykreslování dolní podstavy
endCap	TRUE	Povolení vykreslování horní podstavy
convex	TRUE	Profil je konvexní
ccw	TRUE	Profil je zadán proti směru hodinových ručiček
solid	TRUE	Všechny plochy jsou jednostranné
creaseAngle	0	Mezní úhel pro určení hladkého napojení sousedních ploch

Tabulka 4: parametry uzlu *Extrusion*

Povrchové plošky objektu jsou vytvořeny opláštěním poloh profilů v prostoru a systém vypočte osvětlovací model. Pomocí těchto tří základních parametrů lze modelovat i poměrně složitá tělesa, jak je patrné z následujících obrázků.



Obrázek 5: 3D model pohárku



Obrázek 6: 3D model tkaniny

Na obrázcích 5 a 6 jsou vidět trojrozměrné modely vytvořené pomocí uzlu *Extrusion*. Model kalichu na obrázku 5 byl převzat z práce [1] a model textilní struktury je výstupem programu, který byl vytvořen v rámci této bakalářské práce. Dalším modelem, prezentující možnosti tohoto uzlu, je model kapky, viz obrázek 7.



Obrázek 7: 3D model kapky

Trojrozměrný model kapky byl VRML browserem převeden do grafické podoby ze zdrojového kódu, který je uveden v tabulce 5.

```
#VRML V2.0 utf8

WorldInfo { title "Jan Krivka:drop" }

Viewpoint {
  position      0 2 50
  orientation   0 0 0 0
  description   "Zepredu"
  fieldOfView   0.26
}

Background {skyColor 1 1 1 }

Transform {
  rotation 0 0 0 0
  translation 0 0 0
  children Shape {
    appearance DEF ZLATO Appearance {
      material Material { diffuseColor 0.2 0.2 0.8
                          specularColor 0.8 0.8 0.9
                          transparency 0.500
                        }
    }
  }
}
```

Tabulka 5: výpis textového souboru reprezentující model kapky

```

geometry Extrusion {
  crossSection [ 0 10.000, 1 9.950
                2 9.798, 3 9.539
                4 9.165, 5 8.660
                6 8.000, 7 7.141
                8 6.000, 9 4.359
                10 0.000, 9 -4.359
                8 -6.000, 7 -7.141
                6 -8.000, 5 -8.660
                4 -9.165, 3 -9.539
                2 -9.798, 1 -9.950
                0 -10.000, -1 -9.950
                -2 -9.798, -3 -9.539
                -4 -9.165, -5 -8.660
                -6 -8.000, -7 -7.141
                -8 -6.000, -9 -4.359
                -10 0.000, -9 4.359
                -8 6.000, -7 7.141
                -6 8.000, -5 8.660
                -4 9.165, -3 9.539
                -2 9.798, -1 9.950
                0 10.000 ]

  spine[ 0 2.2 0
        0 2.5 0
        0 3 0
        0 4 0
        0 5 0
        0 6 0
        0 7 0
        0 8 0]

  scale[ 0.08 0.08
        0.13 0.13
        0.17 0.17
        0.19 0.19
        0.17 0.17
        0.1 0.1
        0.05 0.05
        0.02 0.02

]

  solid FALSE
  beginCap TRUE
  endCap TRUE
  creaseAngle 0.78
} } }

```

Tabulka 5: výpis textového souboru reprezentující model kapky - pokračování

4. Nastupující standardy

V současné době je zvýšený zájem o zobrazování objektů v třetím rozměru na internetu. VRML, průkopník v popisu virtuální reality dostupné z internetu, je vytlačován svým nástupcem X3D (popsáno dále). Jako velmi perspektivní se jeví technologie Ajax (popsáno dále), která ve spojení právě s X3D může přinést úplně nové možnosti zobrazování trojrozměrných informací na internetu.

4.1 X3D – Formát pro ukládání 3D scén

Formát X3D (*Extensible 3D*) je standardem organizace ISO (*International Organization for Standardization*) pro trojrozměrnou počítačovou grafiku běžící v reálném čase, označovanou jako virtuální realita. Je nástupcem Virtual Reality Modeling Language (VRML). Má rozšířené možnosti oproti VRML, jedná se například o modelování pomocí NURBS, *Humanoid animation*, *GeoVRML* a další. Umožňuje kódovat scénu pomocí syntaxe XML a má vylepšené programátorské rozhraní (API). X3D definuje několik profilů (rozšiřující sety) pro různé účely, jako je X3D Core, X3D Interchange, X3D CAD, X3D Geospatial a X3D Immersive, a tvůrci webových prohlížečů mohou definovat své vlastní rozšíření ke standardizovanému konsorciu Web3D.

4.2 Ajax – Technologie pro vývoj webových aplikací

AJAX (*Asynchronous JavaScript and XML*) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich opětovného načítání. Ajax ve skutečnosti není konkrétní jedna technologie, ale pojem označující použití několika technologií dohromady s určitým cílem. Největší výhodou je odstranění nutnosti opětovného načítání a překreslování celé stránky při každé operaci, které jsou nutné u klasického modelu WWW stránek. Pokud například uživatel klikne na tlačítko pro udělení hlasu v nějaké anketě, celá stránka se musí znovu načíst ze serveru, třebaže se na ní jen například aktualizují výsledky hlasování a veškerý zbytek obsahu zůstává stejný. Prostřednictvím AJAXu proběhne odeslání hlasu uživatele na pozadí, server zašle jen ty části stránky, které se změnily, a jen tyto části se

uživatelé na stránce aktualizují a překreslí. Uživatel tak má pocit mnohem větší plynulosti práce, která se blíží běžným desktopovým aplikacím. Tato vlastnost značně snižuje zátěž na webové servery a síť obecně. Jelikož není potřeba při každém požadavku sestavit celý HTML dokument, ale pouze provedené změny, je množství vyměňovaných dat výrazně nižší a teoreticky to může mít příznivý vliv i na zátěž databázových serverů a dalších systémů. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů. Tyto aplikace jsou vyvíjeny s využitím technologií:

1.) XHTML a CSS pro prezentaci informací;

(XHTML = *extensible hypertext markup language*, je jazyk pro tvorbu hypertextových dokumentů v prostředí WWW, je následníkem HTML;

CSS = *Cascading Style Sheets*, je jazyk pro popis způsobu zobrazení internetových stránek)

2.) DOM a JavaScript pro zobrazování a dynamické změny prezentovaných informací;

(DOM = *Document Object Model*, je objektově orientovaná reprezentace XML nebo HTML dokumentu)

3.) XMLHttpRequest pro asynchronní výměnu dat s webovým serverem, kde je užíván formát XML, ale je možné použít libovolný jiný formát včetně HTML

AJAX může naopak zvýšit počet vyměňovaných HTTP požadavků, a i když přenášejí nižší množství dat, při nevhodné implementaci zátěž neklesne. Nevýhodou může být, že se stránky chovají jako plnohodnotná aplikace se složitou vnitřní logikou, nikoli jako posloupnost stránek. Problémem AJAXových aplikací také může být síťová latence: potřeba komunikace přes internet má negativní dopady na rychlost odezvy a interaktivitu uživatelského rozhraní.

4.3 Ajax3D – Třetí rozměr na internetu

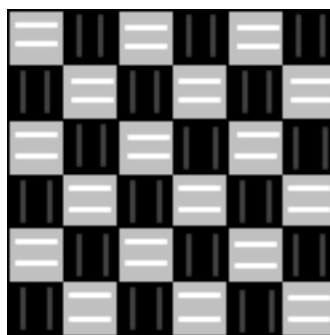
V současné době má třetí rozměr na internetu velkou perspektivu. Firmy, respektive vývojáři, mají možnost pomocí trojrozměrné virtuální scény běžící v reálném čase přinést svým zákazníkům plnohodnotnější informace a doručit je srozumitelným a přirozeným způsobem přímo na internetové stránky [9]. Ajax3D kombinuje přednosti X3D s Ajaxem. Ajax3D používá *X3D Scene Access Interface* (SAI), což je ekvivalentem DOM, který řídí 3D světy pomocí Javascriptu. S použitím X3D zásuvných modulů (*plugin*) mohou dnešní internetové prohlížeče přinášet třetí rozměr každému internetovému uživateli. Ajax3D je postaven na integraci dvou programovacích modelů, na *W3C Document Object Model* (DOM) a na *ISO Scene Access Interface* (SAI). DOM definuje přístup pro programování interakcí s webem a SAI umožňuje programům řídit X3D scénu. Javascript běžící na internetové stránce může používat DOM a SAI simultánně, dovoluje spojit tradiční 2D web s prvky 3D světa. DOM a SAI jsou nezávislé na programovacím jazyce, avšak Ajax3D je zaměřen na Javascript. Ajax3D využívá *server-request* metody (*XMLHttpRequest*, *createX3DFromURL*) k ukládání a získávání dat, aby mohl dynamicky měnit scénu.

5. Základní vazby tkanin

Existují tři základní vazby tkanin, jedná se o plátnovou, keprovou a atlasovou vazbu. Správná volba vazby tkaniny je velmi důležitá [8], protože způsob provázání vláken určuje některé vlastnosti vazby (pevnost, splývavost, tuhost, vzhled, omak aj.).

5.1 Plátnová vazba

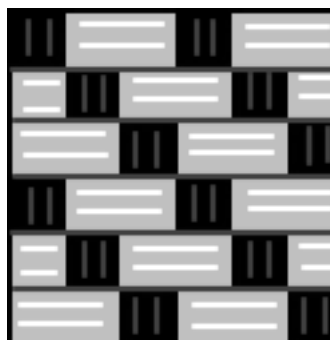
Plátnová vazba je nejpevnější a nejtrvanlivější. Jedná se o nejjednodušší a nejhustěji provázanou vazbu. Střidu vazby tvoří dvě nitě útkové a dvě nitě osnovní. Charakteristickým rysem je pravidelné střídání osnovních a útkových vazných bodů. Plátnová vazba je velmi často používána (košiloviny, ložní a stolní prádlo, plachtoviny, kapesníky aj.). Schematicky je znázorněna na obr. 8.



Obrázek 8: plátnová vazba

5.2 Keprová vazba

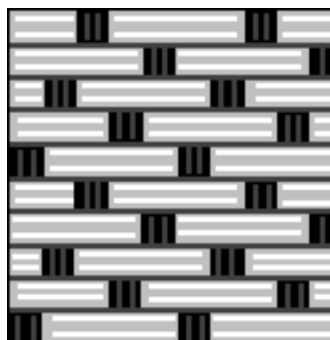
Keprové vazby jsou charakteristické šikmým úhlopříčným (diagonálním) řádkováním levého a pravého směru. Nejmenší střída je 3x3 a vazba může být útková nebo osnovní. Je použita v džínsovině, flanelu aj. Schematicky je na obr. 9.



Obrázek 9: keprová vazba

5.3 Atlasová vazba

Atlasová vazba se vyznačuje svým leskem a šikmým nevýrazným řádkováním. Vazné body jsou pravidelně rozloženy tak, aby se vzájemně nedotýkaly. Atlasová vazba se díky svému lesku používá u dámských společenských tkanin (satén, brokát aj.). Příklad je na obr. 10.



Obrázek 10: atlasová vazba

6. Realizace

Program byl zhotoven v programovacím prostředí Delphi 5.0. Pro grafické rozhraní byly využívány v podstatě čtyři komponenty prostředí Delphi: *Label*, *ComboBox*, *TrackBar* a *WebBrowser*. *Label* je štítek a má informativní charakter, ale byl též použit na funkci tlačítka. *ComboBox* a *TrackBar* jsou komponenty, kterými si lze vybrat nebo přímo nastavit hodnoty parametrů. *WebBrowser* je komponenta pro výstup HTML jazyka a v našem případě slouží k zobrazení nastaveného modelu tkaniny. To, že se může model tkaniny zobrazit v internetovém prohlížeči, je dáno tím, že jazyk VRML je orientovaný na práci na internetu. Nutno dodat, že pro správnou funkci musí být ještě nainstalovaný VRML prohlížeč. Byl vyzkoušen CosmoPlayer a později Cortona VRML Client, viz obrázek 13.

Program na základě nastavených hodnot celé řady parametrů určuje podmínkami (*if-then-else*), které části programu budou vykonány a jakým způsobem budou vykonány. Všechny propočty jednotlivých souřadnic bodů se vykonávají pomocí *for* cyklů. Poté, co program vypočte všechny souřadnice bodů, zapíše výsledný kód do textového souboru s příponou *.wrl*. Strukturu textového souboru je možné vidět na příkladě uvedeném v tabulce 5. Po uložení se tento textový výstup převede pomocí VRML prohlížeče do grafické podoby a zobrazí se.

Úlohou programu je tedy vypočítat hodnoty souřadnic udávající průřez vlákna a profil vlákna, což jsou dva základní parametry uzlu *Extrusion*, který je použit jako základ pro zobrazení. Je možné nastavit další vlastnosti jako je barva, průhlednost, zrcadlivost, nasvětlení a další. Tato nastavení umožňují vytvořit více realistický vzhled a zlepšují vizuální vjem. Dále program propočítává posuny mezi vlákny v závislosti na typu zvolené textilní struktury. Všechny výpočty jsou prováděny na základě matematického popisu textilních vláken struktur. Program automaticky provádí formátování výsledného textového souboru.

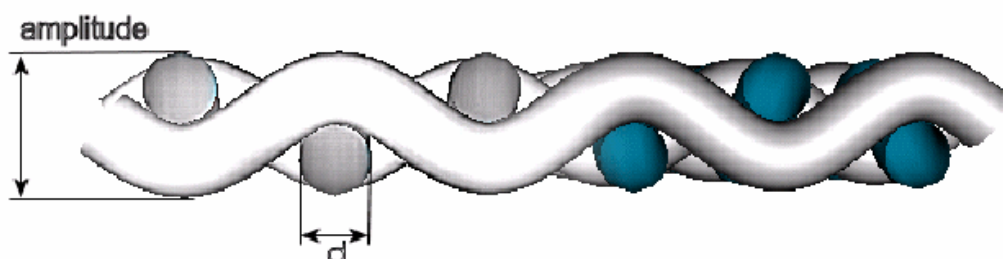
Textilní vlákna byla modelována v idealizované formě. Zhotovený program nabízí možnost vybrat si ze třech základních vazeb tkanin, a to plátňové, keprové a atlasové vazby. Jako tvar průřezu je možné nastavit kruh, elipsu, čtverec nebo obdélník. Hodnoty udávající velikost a tvar průřezu lze nastavit v menu vytvořeného programu a

program sám dopočte konkrétní souřadnice. Předpokládá se, že osa vlákna má sinusový charakter, to znamená, že je dána vztahem

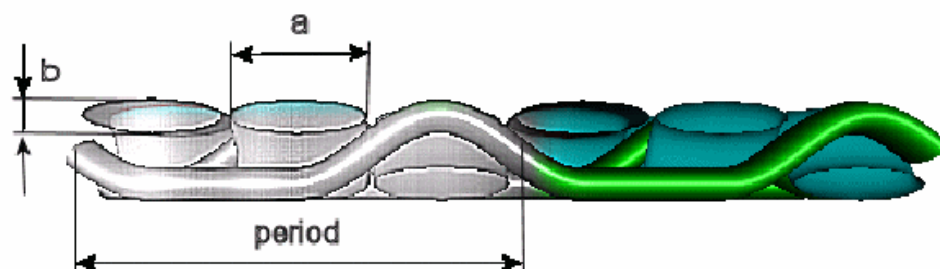
$$y(x) = A \sin\left(2\pi \frac{x}{\lambda} + \varphi\right) \quad (1)$$

kde x je souřadnice osy vlákna, y je pozice bodu na ose, A je amplituda, λ je vlnová délka a φ je počáteční fáze.

Tvar osy vlákna tedy určuje nastavení amplitudy, vlnové délky a počáteční fáze. To platí ovšem pro jednoduchou textilní vazbu, viz obrázek 11. V případě více komplikovaných struktur jako je kepr nebo atlas, jsou části sinusovek střídány rovnými částmi, viz obrázek 12. Počáteční fáze je kritickým parametrem pro jednotlivá modelovaná vlákna, poněvadž musí zajistit, aby se právě dotýkala, ale neprotínala, či neprobíhala mimo. Algoritmus pro správný model vazby je tedy dosti komplikovaný. Detaily z důvodu stanoveného rozsahu práce neuvádíme.

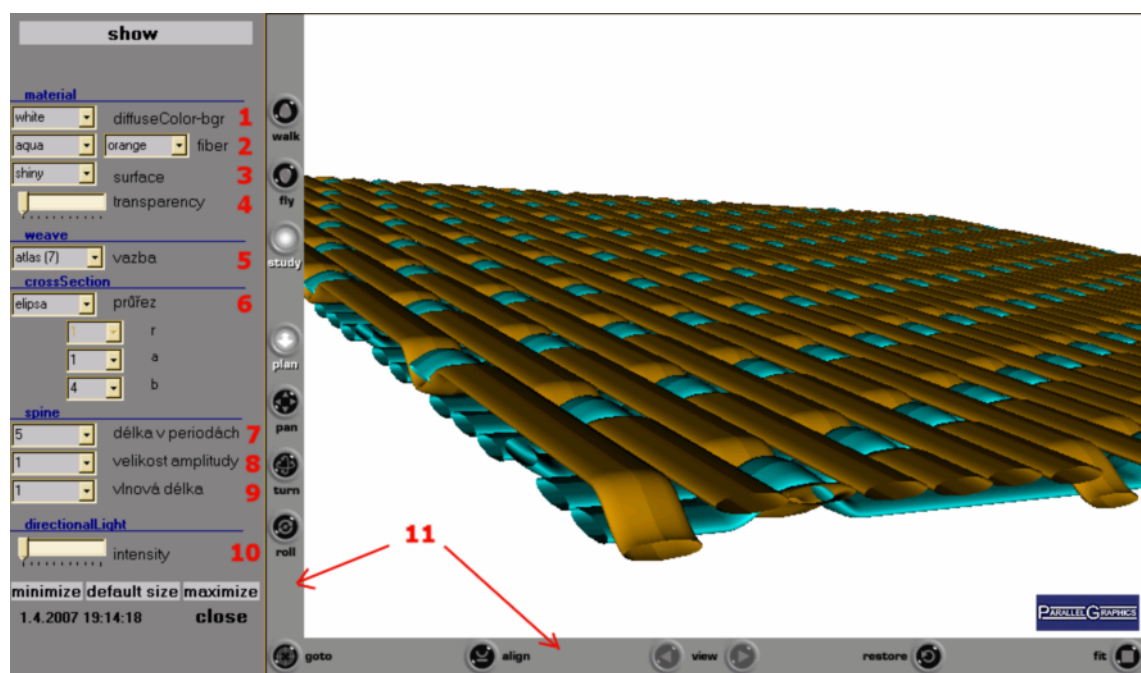


Obrázek 11: model plátna s kruhovým průřezem vláken



Obrázek 12: model kepru s průřezem vláken ve tvaru elipsy

Program je ovládán z menu, a tím umožňuje použití pro nezasvěcené uživatele, viz obrázek 13 a tabulka 6. Program v podstatě okamžitě po změně vlastností textilní struktury zajistí její zobrazení do grafické podoby, běží tedy téměř v reálném čase.



Obrázek 13: grafické rozhraní s popisem zhotoveného programu

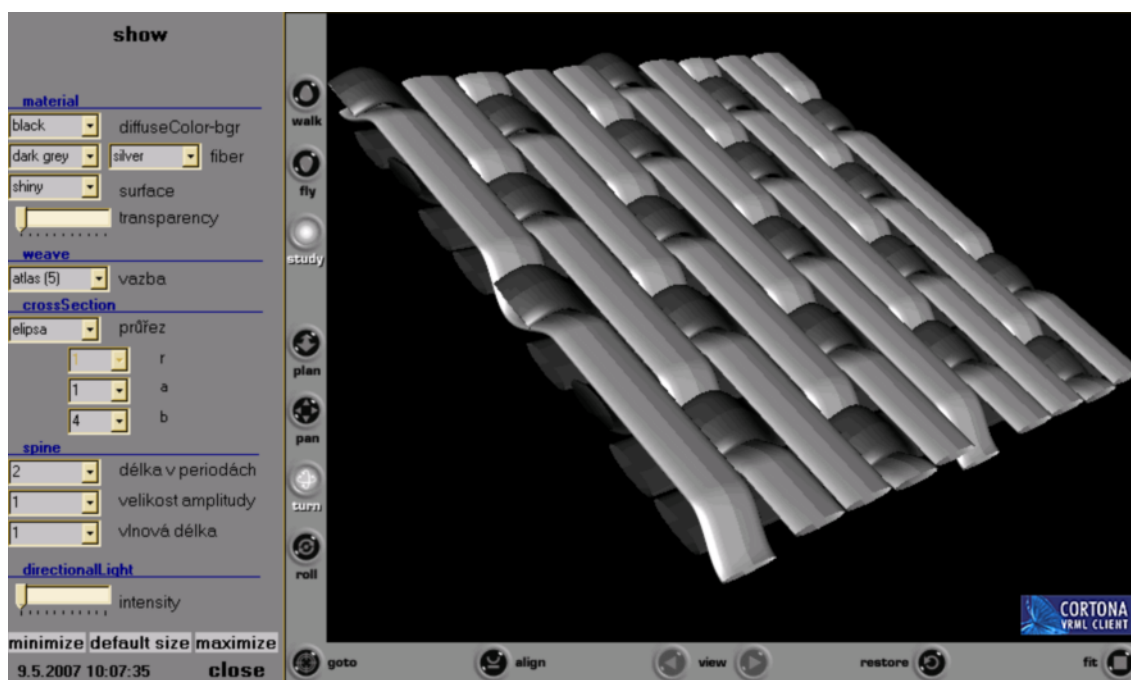
	Popis možností nastavení	Aktuální nastavení podle obrázku 13
1	Nastavuje barvu pozadí (20 přednastavených barev)	white – bílé pozadí
2	Nastavuje barvu příčných a podélných vláken	aqua, orange – akvamarínová a oranžová barva vláken
3	Nastavuje matnost resp. lesklost povrchu	shiny – lesklý povrch
4	Nastavuje průhlednost	transparency – průhlednost (vypnuto)
5	Nastavuje typ vazby	sedmivazný atlas
6	Nastavuje tvar a velikost průřezu	elipsa 1:4
7	Nastavuje velikost tkaniny	5 buněk po 7 vláknech, tj. 35*35 vláken
8	Nastavuje velikost amplitudy	1
9	Nastavuje vlnovou délku	1
10	Nastavuje velikost přímého osvětlení	0
11	Ovládací lišty VRML prohlížeče (v tomto případě Cortona)	

Tabulka 6: Volitelné parametry zhotoveného programu. Indexování je podle obr. 13.

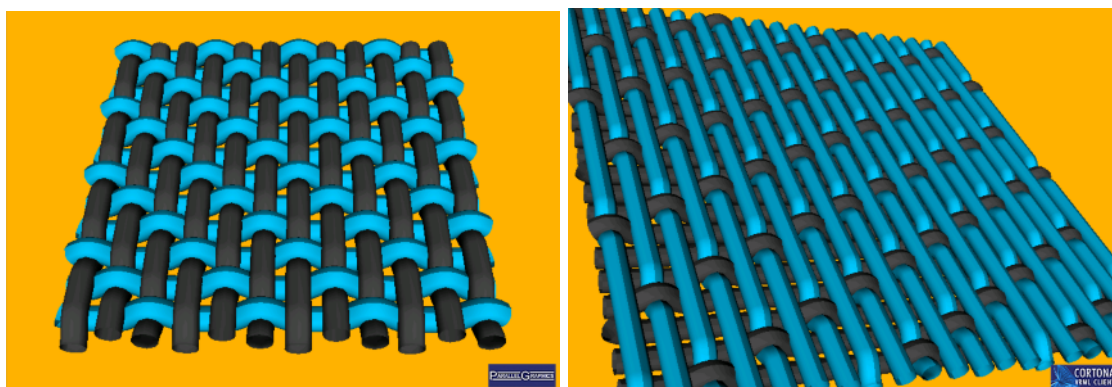
VRML prohlížeč nabízí interakci s grafickým prostředím. Jedná se o pohyb virtuálním světem a manipulaci s modelem, umožňuje procházení strukturou, plynulý přesun na přednastavené pozice náhledu aj.

7. Typické výsledky

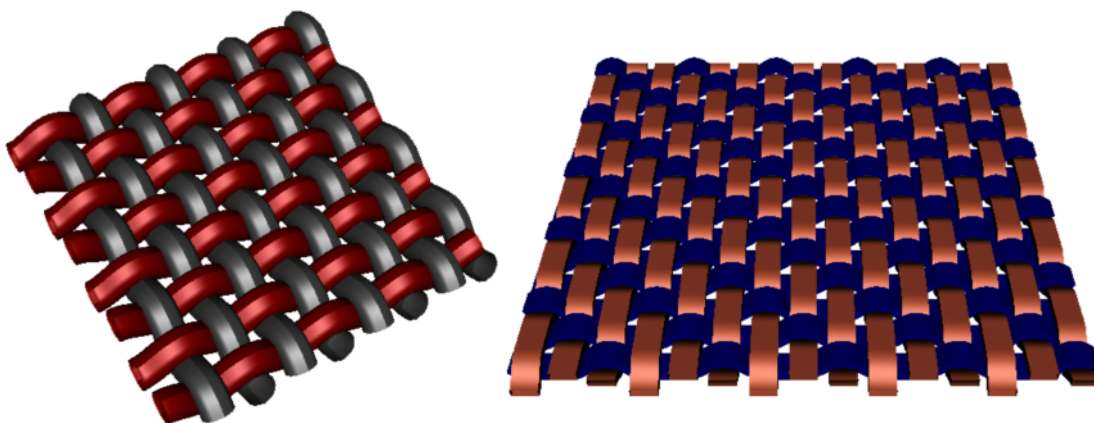
Výsledky demonstrují následující obrázky modelů různých textilních vazeb. Všechny tyto modely byly vytvořeny programem napsaným v rámci této bakalářské práce. Na obrázku 14 je pohled na program s modelem tkaniny, na dalších obrázcích jsou jen samotné modely, ale byly vytvořeny tím samým programem:



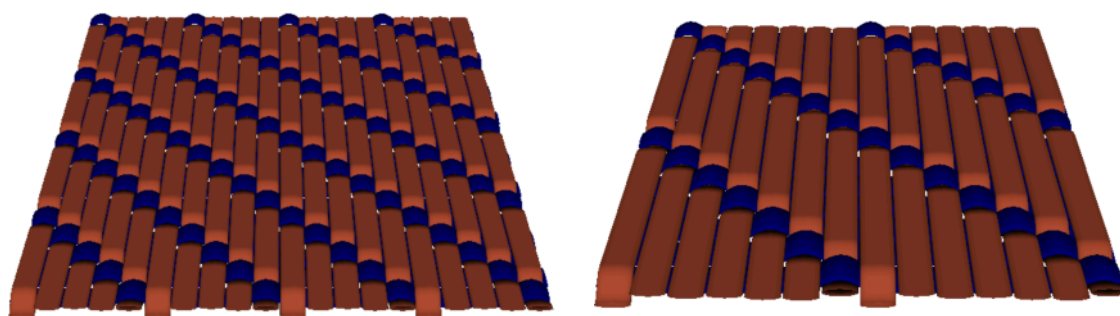
Obrázek 14: výsledný program zobrazující model tkaniny



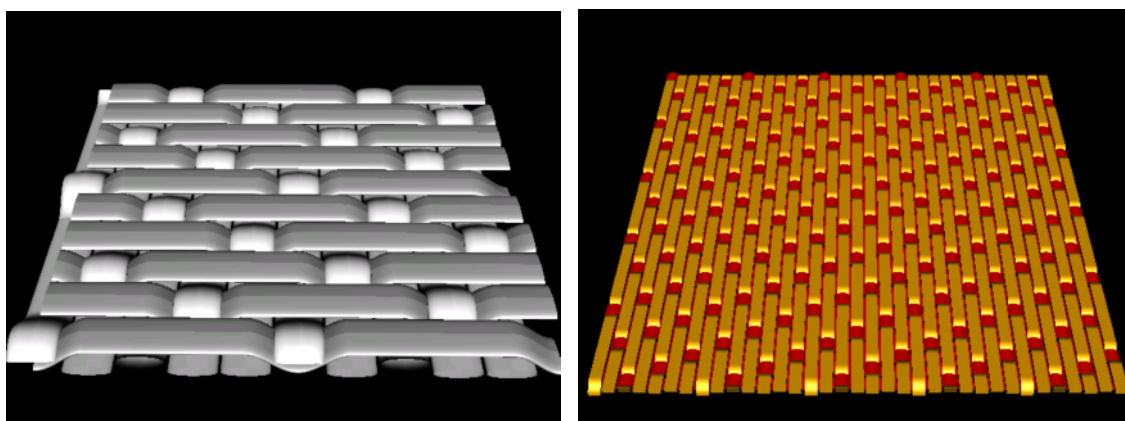
Obrázek 15: 3D model třívazného kepru s kruhovým průřezem vláken a 3D model sedmivazného atlasu s kruhovým průřezem vláken



Obrázek 16: 3D model plátka s kruhovým průřezem vláken a 3D model třívazného kepru s obdélníkovým průřezem



Obrázek 17: 3D model pětivazebného kepru s obdélníkovým průřezem vláken a 3D model sedmivazebného kepru s eliptickým průřezem



Obrázek 18: 3D model pětivazebného atlasu s eliptickým průřezem vláken a 3D model sedmivazebného atlasu s eliptickým průřezem

8. Diskuse

Modelování textilních struktur se řeší na TUL poměrně dlouhou dobu, během níž byly vypracovány modely různé úrovně složitosti. V tomto výčtu jsou seřazeny podle rostoucí složitosti:

1. Aproximace vláken pomocí elementárních válců [4].
2. Aplikace uzlu *Extrusion*.
3. Aplikace povrchových plošek pomocí VRML [5,6].
4. Aplikace povrchových plošek na základě OpenGL [7].

Elementární válce [4] neposkytovaly mnoho možností a proto se od vývoje brzy upustilo. Typické výstupy uzlu *Extrusion* jsou v této práci. Aplikace povrchových plošek pomocí VRML byla řešena ve spolupráci s ústavem MTM na Katholieke universiteit Leuven. Zobrazovací část původního velmi rozsáhlého návrhového systému WiseTex byla převedena do VRML. Typické výsledky jsou v příloze 1 [5,6]. Podobně jsou v příloze 2 výsledky aplikace jiného systému založeného na OpenGL [7], jehož úkolem je sledovat vliv vad v textilních strukturách.

V principu dává uzel *Extrusion* tytéž možnosti jako složitější systémy. Při vhodné volbě tvaru průřezu vláken lze tedy dostat výstupy stejné úrovně jako jsou v přílohách od podstatně složitějších systémů. Předností zvolené metody je zejména to, že uzel *Extrusion* dává model struktury v reálném čase, takže by bylo např. možné animovat plynulou změnu jednoho nebo i více parametrů struktury.

Závěr

Práce ukázala, že v případě modelů základních textilních struktur lze získat velmi jednoduchou metodou téměř stejné výsledky, jaké poskytují složité systémy. Hlavní výhodou použité metody je její jednoduchost. Uživatel se může zaměřit na řešení problému, jako popis struktury. Výpočty a převedení do trojrozměrné grafické podoby provede samotný program. Tato výhoda je limitována složitostí modelů, při větší složitosti modelů dochází ke zpomalení zobrazení modelu. Z praktického hlediska tyto limity nejsou tolik zásadní.

Hlavním přínosem systému je dále to, že i nezkušený uživatel může jednoduše vizualizovat různé modely textilních struktur a výsledky jsou srovnatelné s komerčními systémy. Program v porovnání s obdobnými programy není hardwarově náročný, respektive nevyžaduje vysoký výpočetní výkon procesoru. Kromě programu samotného a VRML prohlížeče, nemusí mít uživatel nainstalované žádné další programy. Výsledný kód, který je uložen v textovém souboru, je čitelný a stručný.

Program umožňuje interaktivním způsobem sledovat vliv všech parametrů vazby a vláken na výslednou strukturu modelu textilie. Základní možnosti ukázaly ilustrativní obrázky v části 7 – Typické výsledky. Model textilie může být při vhodném použití barev a textur blízky realitě. Velkou výhodou je i to, že může být prohlížen ze všech stran a vzdáleností. Volba vhodných parametrů a možnost důkladného prohlížení umožní optimalizovat počítačový návrh. Při troše zkušeností by tento systém mohl zkrátit nebo přímo nahradit návrh nového typu textilie. Program a jím vytvořené 3D modely mohou být použity ve vzdělávání, v průmyslu, pro potřeby reklamy a určitě i v jiných technických oblastech.

Podle zadání byly v plném rozsahu splněny první dva body. Třetí bod nebyl řešen, poněvadž se nepodařilo získat dostatečně kvalitní řezy strukturou. Dále měly být výsledné modely převedeny do skutečného 3D světa pomocí stereoskopického hardwaru. Po SW stránce bylo vše připraveno. Podařilo se též zakoupit kvalitní monitor. Byly objednány brýle pro virtuální realitu od firmy eDimensional, ale po komplikacích s dodavatelskou firmou, která objednávku neprovedla, bylo od záměru upuštěno.

Díličí výsledky práce byly publikovány na textilní konferenci s mezinárodní účastí v Ružomberoku, srpen 2006 [10].

Použitá literatura

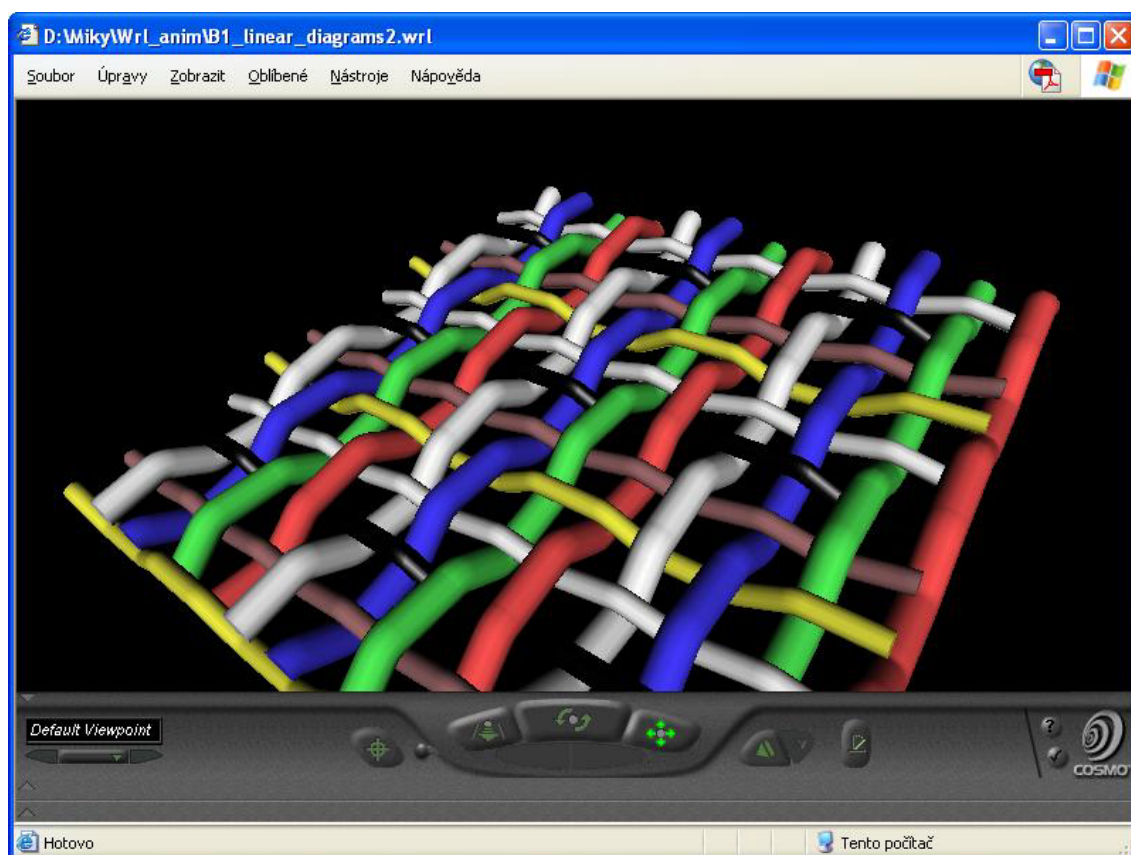
- [1] Jiří Žára *VRML 97 Laskavý průvodce virtuálními světy*. Vydavatelství Computer Press, Praha 1999, ISBN 80-7226-143-6
- [2] Jiří Žára, Bedřich Beneš, Petr Felkel *Moderní počítačová grafika*. Vydavatelství Computer Press, Praha 1998, ISBN 80-7226-049-9
- [3] Jakub Zrzavý *VRML tvorba dokonalých WWW stránek*. Vydavatelství Grada Publishing, Praha 1999, ISBN 80-7169-643-9
- [4] Košek, M.: *Virtual Reality – an Efficient Tool for Visualisation and Modelling Textile Structures*, Proceedings of Spring Conference on Computer Graphics, SCCG 2001, Conference Materials, Posters, Budmerice, Slovak Republic, April 25th – 29th, 2001, str. 31-32.
- [5] Mikolanda T.: *VRTex - Virtual Reality Textiles* [online].2004. [cit. 08.05.2007]
Dostupné na www: <<http://centrum.vslib.cz/vrtex>>
- [6] Mikolanda T., Lomov S. V., Košek M.: *VRTex – an Effective System for Visualization of Textile Structures*, STRUTEX 2003 – 10th international Conference on Structure and Structural Mechanics of Textile Fabrics, December 8-9 2003, Technical University of Liberec, Liberec, Czech Republic, str. 33-38.
- [7] Tunák M.: *3D vazby* [online].2006. [cit. 08.05.2007]
Dostupné na www: <<http://www.ft.vslib.cz/depart/ktm/?q=cs/vazby>>
- [8] Dostalová M., Křivánková M.: *Základy textilní a oděvní výroby*, Vydáno Technickou univerzitou v Liberci, Liberec 2004, ISBN 80-7083-831-0
- [9] Parisi T.: *Ajax3D: The Open Platform for Rich 3D Web Applications* [online].2006. [cit. 08.05.2007]
Dostupné na www: <<http://www.ajax3d.org/whitepaper/>>

Přílohy

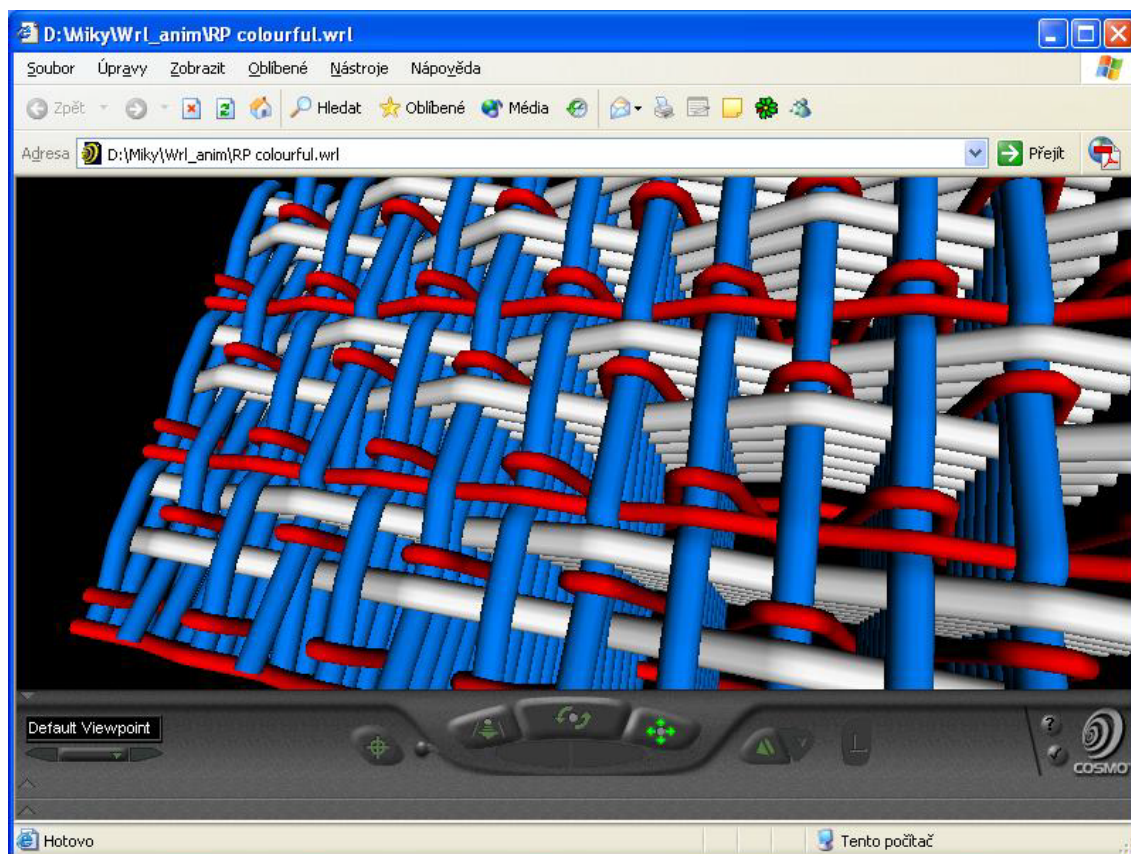
Příloha 1: Projekty obdobného zaměření - Práce Ing. Tomáše Mikolandy

Dostupné na:

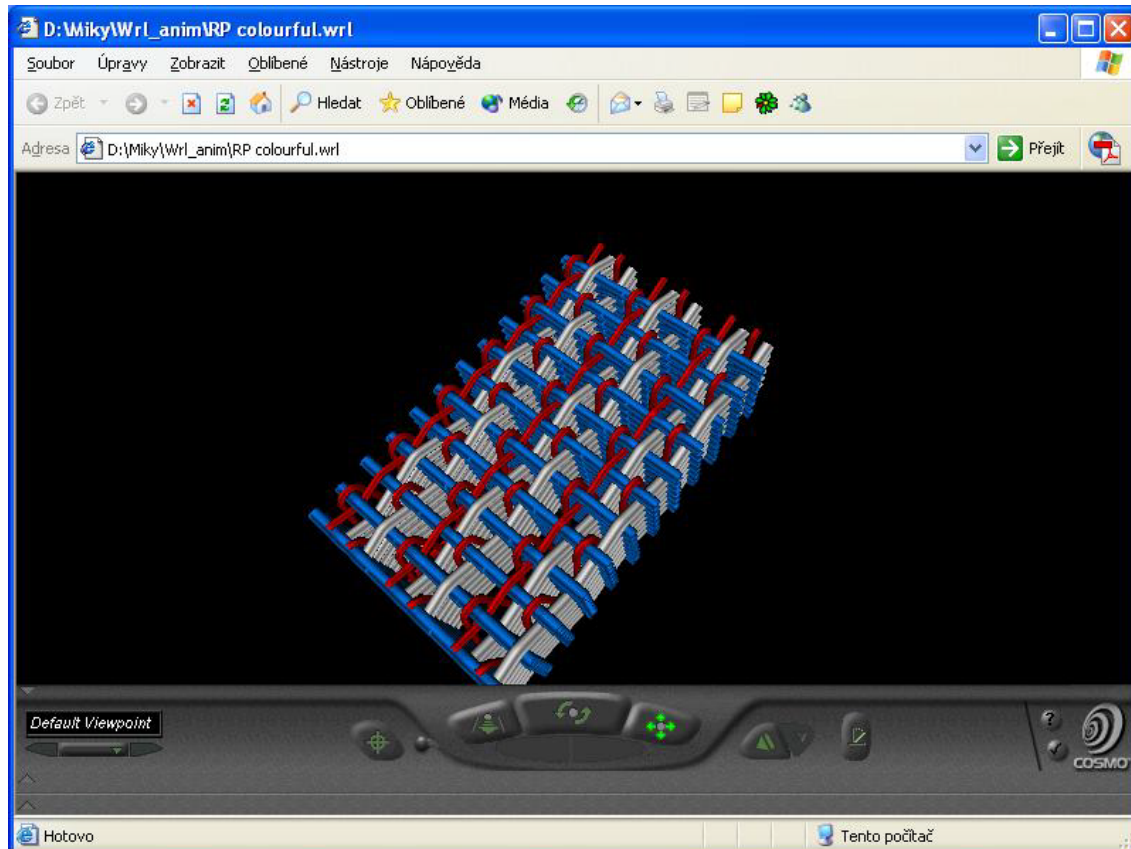
<http://centrum.vslib.cz/vrtex>



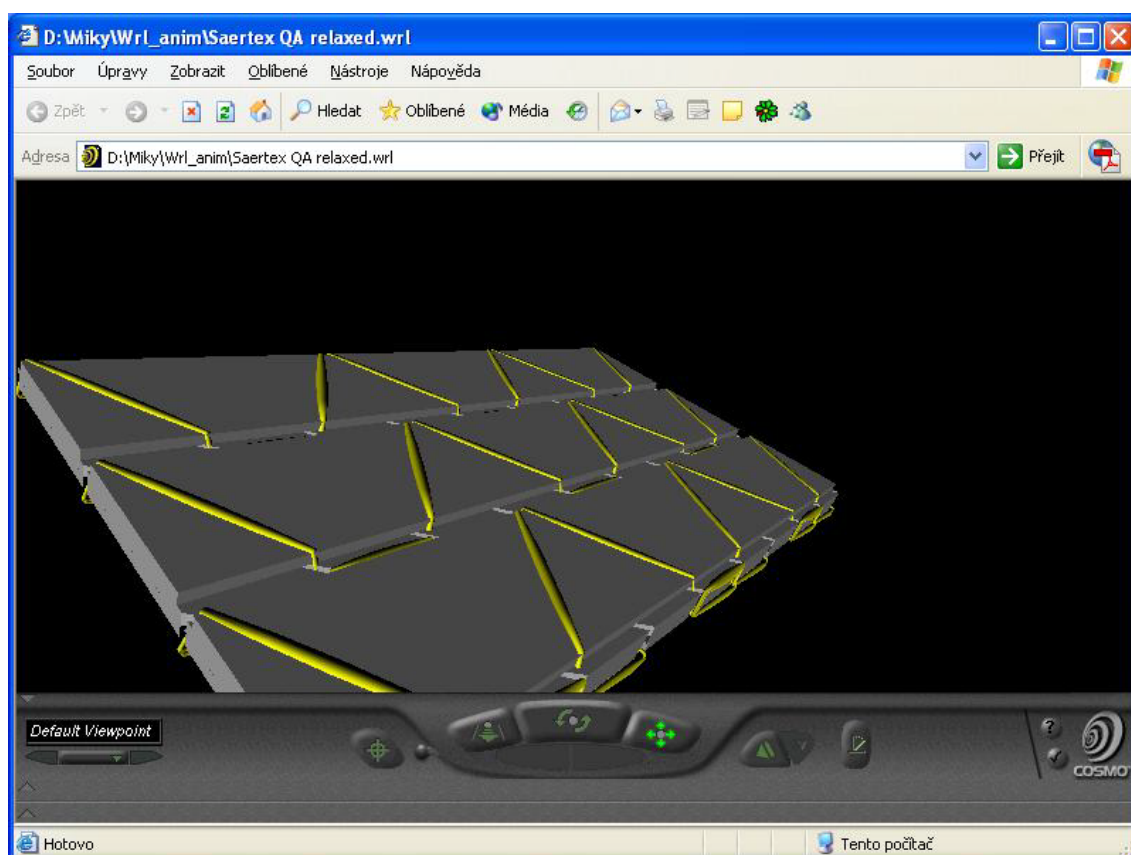
Obrázek 19: model tkaniny 1



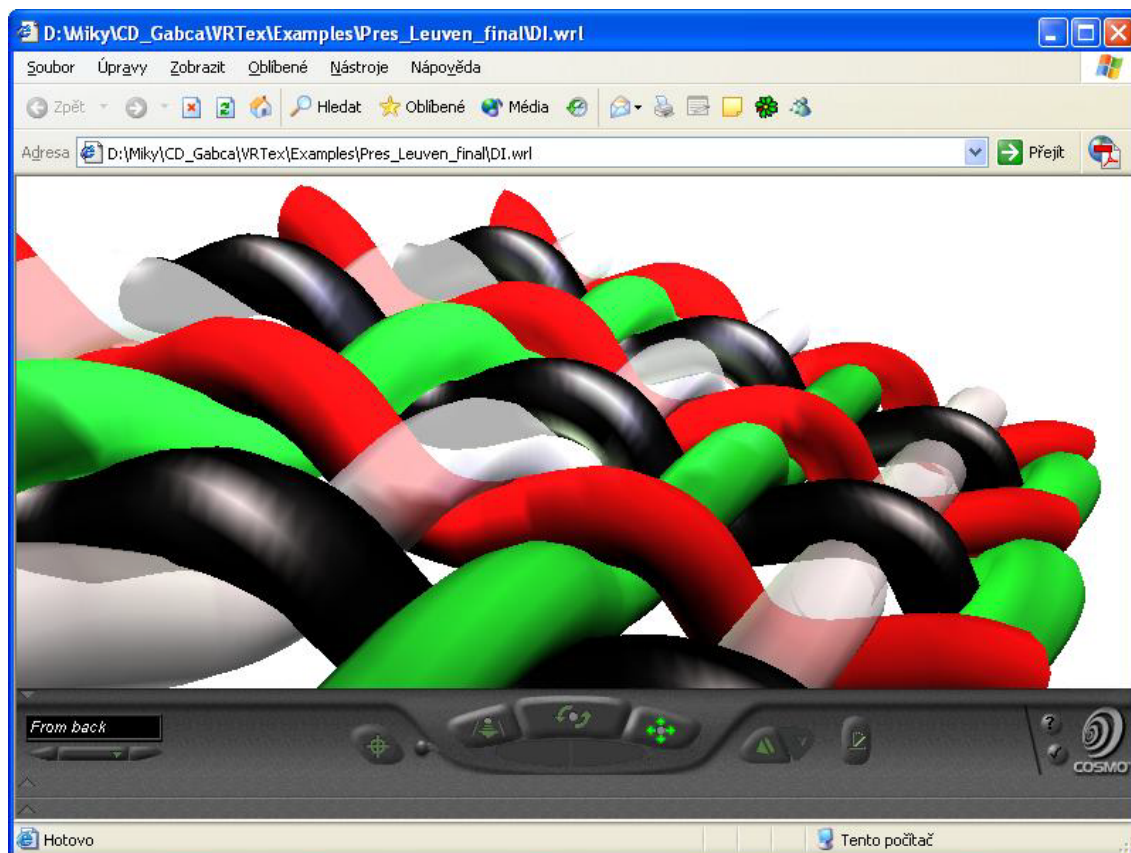
Obrázek 20: model tkaniny 2



Obrázek 21: model tkaniny 3



Obrázek 22: model tkaniny 4

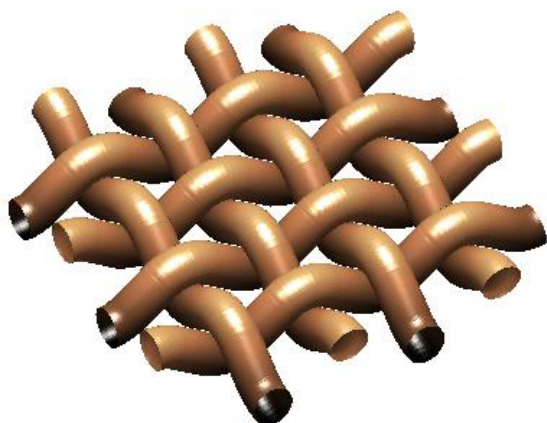


Obrázek 23: model tkaniny 5

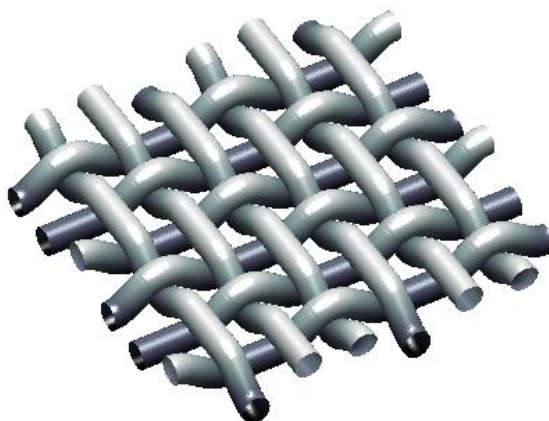
Příloha 2: Projekty obdobného zaměření - Práce Ing. Maroše Tunáka

Dostupné na:

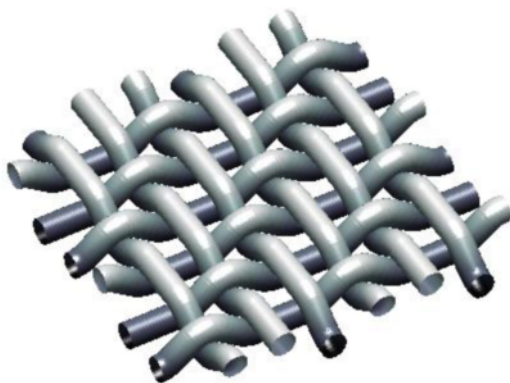
<http://www.ft.vslib.cz/depart/ktm/?q=cs/vazby>



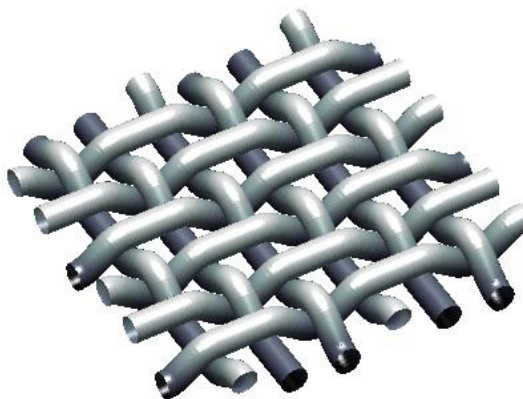
Obrázek 24: plátnová vazba



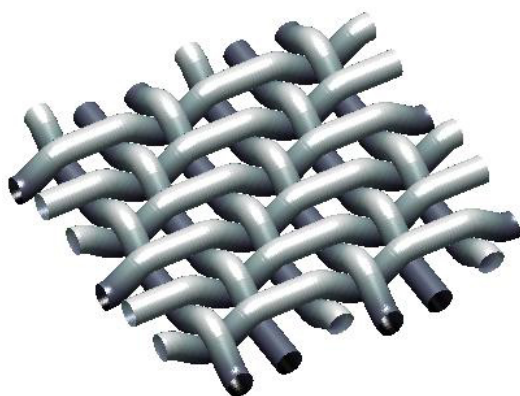
Obrázek 25: třívazný osnovní kepr
levého směru



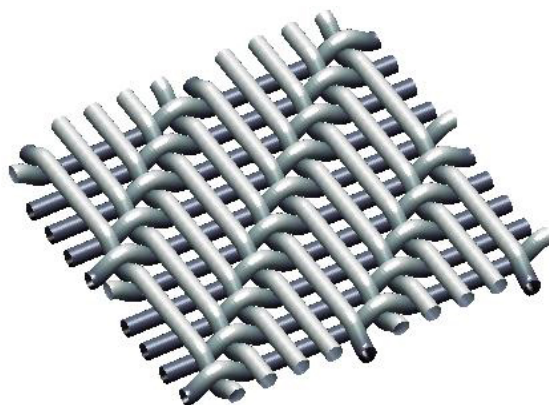
Obrázek 26: třívazný osnovní kepr pravého
směru



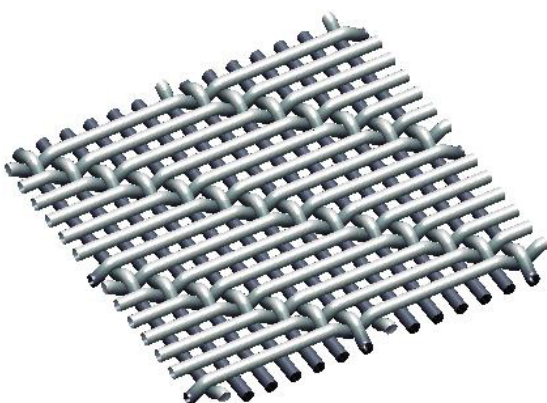
Obrázek 27: třívazný útkový kepr
levého směru



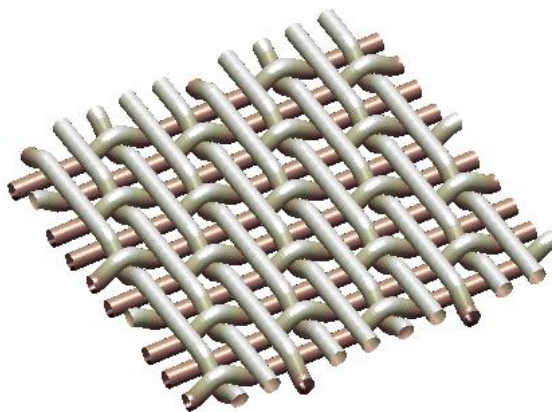
Obrázek 28: třívazný útkový kepr pravého směru



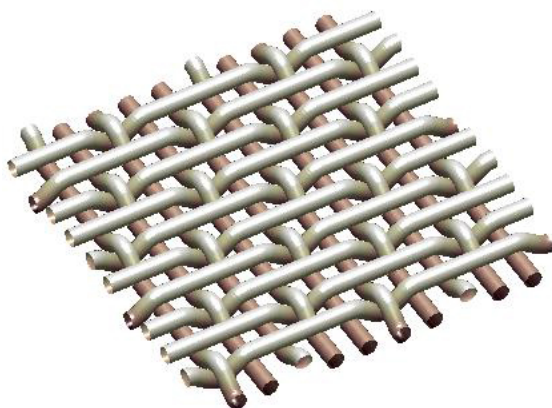
Obrázek 29: pětivazný osnovní kepr pravého směru



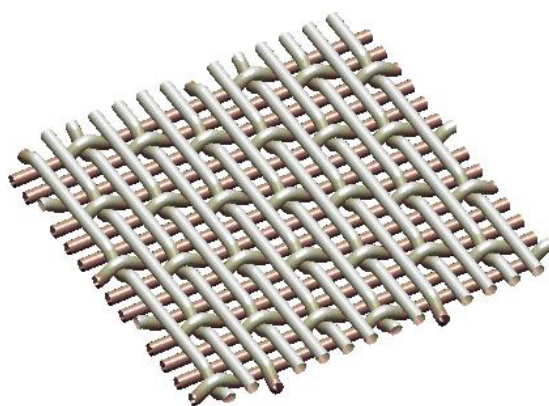
Obrázek 30: sedmivazný útkový kepr levého směru



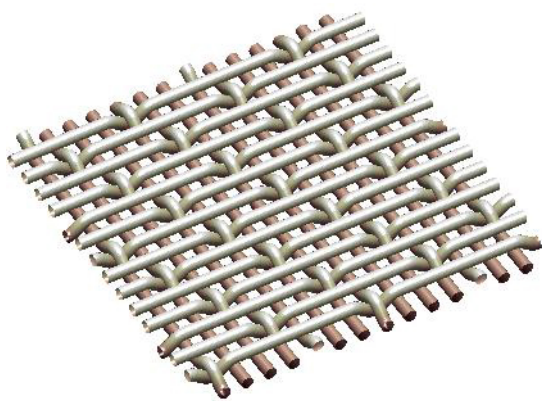
Obrázek 31: pětivazný osnovní atlas



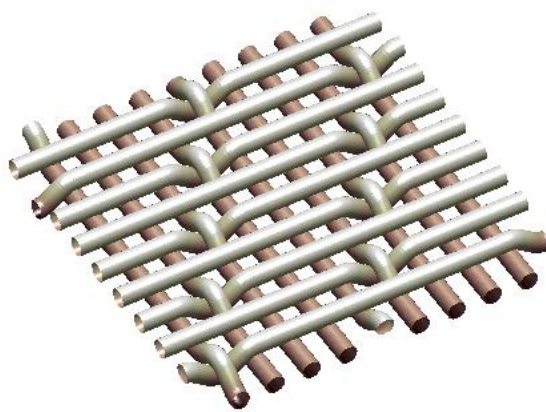
Obrázek 32: pětivazný útkový atlas



Obrázek 33: sedmivazný osnovní atlas



Obrázek 34: sedmivazný útkový atlas



Obrázek 35: devítivazný útkový atlas